

**Mobile und Verteilte Systeme
Ubiquitous Computing
Teil VI
Seminar im WS 2008/09**

**Herausgeber: Till Riedel, Martin Berchtold,
Christian Decker**

Interner Bericht 2009-3



Universität Karlsruhe (TH)
Forschungsuniversität · gegründet 1825

ISSN 1432-7864



Fakultät für **Informatik**

Vorwort

Das Seminar „Mobile und verteilte Systeme – Ubiquitous Computing“ am Telecooperation Office (TecO) erfreut sich großer Beliebtheit. Der vorliegende interne Bericht enthält Beiträge von Studierenden zu diesem Seminar, das im WS 2008/09 in dieser Form zum sechsten Mal stattgefunden hat.

Die Themenauswahl für das Seminar orientiert sich im wesentlichen an aktuellen wissenschaftlichen Fragestellungen in den Bereichen:

- Sensorverarbeitung und Kontexterkenkung
- Lokalisierung und Navigation
- Modelle für ubiquitären Informationssystemen

Aufgrund des stetig wachsenden Interesses der Studierenden an diesen Themenbereichen haben wir uns entschlossen einen Seminarband mit den Beiträgen unserer Studierenden als internen Bericht zu veröffentlichen. Durch die engagierte Mitarbeit der beteiligten Studierenden wird ein Ausschnitt aus diesem komplexen und umfassenden Themengebiet klar und übersichtlich präsentiert. Für den Fleiß und das Engagement unserer Seminarteilnehmer wollen wir uns an dieser Stelle daher herzlich bedanken.

Bestärkt durch die gute Resonanz der Studierenden, werden wir dieses Seminar auch im nächsten Wintersemester – dann natürlich mit aktualisierten Themen – wieder anbieten.

Karlsruhe, Februar 2009

Till Riedel, Martin Berchtold, Christian Decker

Inhaltsverzeichnis

Mobile und verteilte Systeme	1
<i>Till Riedel, Martin Berchtold, Christian Decker (Hrsg)</i>	
Machine Health Monitoring	17
<i>Dilek Fansa</i>	
Verarbeitung und Erzeugung von ungewissen und unscharfen Informationen	31
<i>Nadia Ahmed</i>	
Simulation von Ubiquitären Systemen	45
<i>Adrian Genaid</i>	
Formale Modelle für Ubiquitäre Informationstechnologien	63
<i>David Lisowski</i>	
Probabilistisches Model Checking	79
<i>Tobias Cichos</i>	
Simultaneous Localization and Mapping	95
<i>Thomas Lörcher</i>	
Unterstützende Technologien für den Einsatz in Gefahrensituationen	113
<i>Wolfgang Heetfeld</i>	

Machine Health Monitoring

Dilek Fansa

Universität Karlsruhe (TH)

1 Machine Health Monitoring

1.1 Motivation

Ubiquitäre Informationstechnologien sind in den elektronischen Sensoren weit verbreitet. Sensoren werden in vielen Bereichen wie Vibration-monitoring, Temperatur Messungen, Gassensing, Bildverarbeitung etc. verwendet. Die Maschinenvibration enthält fast alle Informationen über den Anlagenzustand, weshalb die Schwingungsüberwachung häufig für die frühzeitige Detektion von Maschinenschäden im Rahmen von Machine Health Monitoring (Zustandsüberwachung) eingesetzt wird. Die Sensoren messen die physikalische Größe von Schwingungen und wandeln sie in elektronische Signale um, die in andere Geräte verarbeitet und übertragen werden können. Die Online-Zustandsüberwachung ist eine der zuverlässigsten und auch bequemsten Methoden um Schäden frühzeitig zu erkennen. Allerdings sind die Anschaffungskosten von Online-Systemen meist hoch. Neue Technologien wie miniaturisierte Sensoren und drahtlose Übertragung ermöglichen flexible Maschinendiagnose. Der Sensoren, der im Bereich Machine Health Monitoring verwendet wird, kann ein Beschleunigungssensor, Acoustic Emission Sensor, Thermocouples oder Mikrofon sein.

Das Konzept des Machine Health Monitoring basiert auf einer regelmäßigen Erfassung des Maschinenzustandes durch Messung aussagefähiger Größen (auch als Zustandsüberwachung bezeichnet). Zustandsüberwachung von Maschinen und Fehlerdiagnose aus dem Schwingungsbild haben eine starke und nach wie vor stark wachsende Bedeutung im industriellen Umfeld. Die meisten Ausfälle werden kurz vor ihrem Auftreten durch eine Warnung gemeldet. Diese Warnung wird als potenzieller Ausfall bezeichnet und als feststellbarer Zustand definiert, der anzeigt, dass eine Funktionsstörung kurz bevorsteht oder bereits vorliegt. Die Zustandsüberwachung hat zum Ziel, Informationen bereitzustellen, die den Betrieb der Maschine länger und mit möglichst geringen Kosten aufrecht erhält. Die Zustandsüberwachung und die vorausschauende Wartung öffnen der Instandhaltung die Tür von einer aufwendigen, nie endenden Notfallbekämpfung zu einem effizienten, organisierten Prozess mit gezielten und kostenbewussten Maßnahmen. Bei der Maschinenüberwachung, die den Schutz der Maschine und ihrer Komponenten zum Ziel hat, ist das Verhalten der Maschinenstruktur die Signalquelle. Treten Eigenschaftsänderungen in der Maschinenstruktur auf, z. B. durch Verschleiß von Bauteilen, so stellt dies für die Maschinenüberwachung ein zu detektierendes Ereignis dar.

1.2 Das Prinzip der Zustandsüberwachung von Maschinen

- Erkennung von Zustandsveränderungen einer Maschine, die zu einer Störung führen können.
- Festlegung physischer Eigenschaften, die gemeinsam den aktuellen Zustand der Maschine anzeigen.
- Messung, Analyse und Aufzeichnung dieser Eigenschaften, um Trends erkennen zu können.
- Über einen längeren Zeitraum hinweg zeigen diese Trends die Verschlechterung des Maschinenzustands an. Sie können für die Festlegung entsprechender Wartungsmaßnahmen genutzt werden.

Teilschritte der Zustandsüberwachung Die Zustandsüberwachung setzt sich aus mehreren Teilschritten zusammen:

1. Zustandserfassung: Die Zustandserfassung ist die Messung und Dokumentation von Maschinenparametern, die den aktuellen Zustand des Produktionsmittels (oder des Bearbeitungsprozesses) widerspiegeln.
2. Zustandsvergleich: Der Zustandsvergleich stellt den Vergleich des Istzustandes mit einem vorgegebenen Referenzwert dar. Dieser Referenzwert kann sowohl ein einzuhaltender Sollwert als auch ein nicht zu überschreitender Grenzwert sein. Der Sollwert wird je nach untersuchtem Parameter entweder bei der Maschinenabnahme ermittelt oder durch vorgegebene Größen festgelegt. Grenzwerte werden meist vom Hersteller oder Anwender der Maschine empirisch ermittelt.
3. Diagnose: Es ist die Aufgabe der Diagnose anhand der Resultate des Zustandsvergleichs eventuell vorhandene Fehler möglichst früh zu lokalisieren und deren Ursache(n) zu ermitteln, um notwendige Instandhaltungsmaßnahmen beizeiten planen zu können.

Die steigenden Kosten im Zusammenhang mit Maschinen, Reparatur- und Ausfallzeiten hat eine Nachfrage nach höheren Wartungs-Techniken

1.3 Strategien in der Wartung

In der heutigen Instandhaltung werden unterschiedliche Strategien genutzt.[1]
Die wichtigsten Strategien sind dabei:

Reaktiven Wartung(RM) Reaktive Wartung ist im Grunde die "run till it breaks" Wartungs-Modus. Bei der Reparatur im Schadensfall (RM) bleibt ein Bauteil solange im Einsatz, bis es ausfällt. Dies ist immer dann sinnvoll, wenn das Bauteil nicht viel kostet, es leicht beschafft werden kann, der Instandhaltungsaufwand gering ist und wenn die Instandsetzungszeit den Produktionsablauf nicht sonderlich stört. Ein typisches Beispiel ist der Tausch von (unkritischen) Lampen nach Defekt.[2]

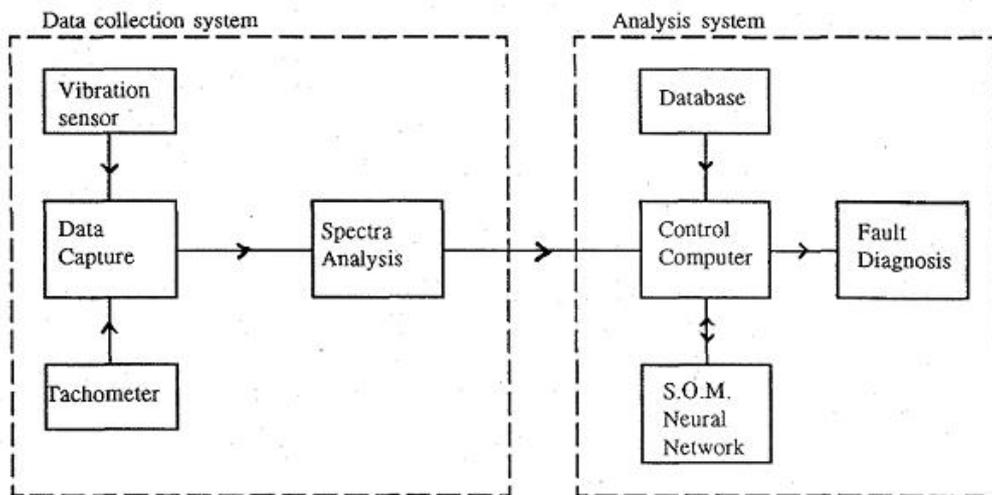


Abbildung 1. Diagramm der Zustandsüberwachung

- Vorteile
 - Geringe Kosten.
 - Weniger Personal.
- Nachteile
 - Höhere Kosten durch ungeplante Ausfallzeiten der Ausrüstung
 - Erhöhung der Lohnkosten, insbesondere durch Überstunden.
 - Höhere Kosten mit der Reparatur oder dem Austausch der Ausrüstung.
 - Mögliche sekundäre Ausrüstung oder Prozess-Schäden durch Ausfall von Geräten.
 - Ineffiziente Nutzung der personellen Ressourcen.

Vorbeugende Wartung (PM) Es werden vorbeugende Maßnahmen (wie Inspektionen und Wartungen) durchgeführt, um ggf. vor Auftritt eines Fehlers Maßnahmen zu ergreifen. Diese Aktivitäten können zum einen zeitbasiert sein. Es werden jedoch auch Strategien genutzt, die auf anderen Kriterien (wie Laufzeiten, Stückzahlen) basieren. Ein typisches Beispiel ist der Ölwechsel in festen Intervallen oder nach einer bestimmten km-Leistung im Kfz. Durch einfaches verwenden wird die Lebensdauer der Ausrüstung verlängert und die Zuverlässigkeit erhöht. Damit ermöglicht PM eine Senkung von Kosten, Energieaufwand und Ausfall von Prozess und Komponenten. Circa 12% bis 18% Kostenersparnis gegenüber RM.

Katastrophale Ausfälle von Geräten sind nicht zu verhindern. Inklusive Leistung unbenötigter Wartung für zufällige Schäden an Komponenten in Durchführung ist immer möglich.[2]

- Vorteile

- Kostengünstig in vielen kapitalintensive Prozesse.
- Erhöhung der Zuverlässigkeit
- Erhöhung der Lebensdauer der Komponenten .
- Energie-Einsparungen.
- Reduzierte Ausrüstung oder Prozessausfall.
- Geschätzte 12
- Nachteile
 - Katastrophale Fehler treten wahrscheinlich auf.
 - Labor intensiv.
 - Inklusive Leistung unbenötigter Wartung.
 - unbenötigte Wartung für zufällige Schäden an Komponenten in Durchführung .

Vorausschauende Wartung(PDM) Es werden auf Basis von vorliegenden Informationen (z.B. Verschleiß) die notwendigen Zeitpunkte zur Durchführung von notwendigen Wartungen bestimmt. Mittels der vorausschauenden Instandhaltung wird der Zustand eines Bauteils während des Produktionsbetriebes überprüft. Bei signifikanten Zustandsänderungen wird dieses Bauteil während eines geplanten Stillstandes gewechselt. Für eine vorausschauende Instandhaltung sind zwei Dinge wichtig:

- Es muss eine geeignete Messmethode verfügbar sein, um Zustandsänderungen eines Bauteils ermitteln zu können.
- Ist ein schadhaftes Bauteil erkannt worden, muss noch genügend Zeit vorhanden sein, um das Bauteil während eines geplanten Stillstandes wechseln zu können.

Seit der Einführung von PDM ergeben sich jährlich beträchtliche Einsparungen durch vermiedene Ausfallzeiten an kritischen Produktionsanlagen, Reduzierung von Material - und Lohnkosten sowie einer besseren Verfügbarkeit der entsprechenden Anlagen. Außerdem führt PDM zu einer verbesserten Planbarkeit von Instandhaltungsaufgaben, d.h. Instandhaltungsarbeiten werden bei objektivem Bedarf ausgeführt. Zusätzlich können Wartungsarbeiten soweit wie möglich hinausgezögert werden, ohne einen Produktionsausfall zu riskieren. Systeme bieten dem Analytiker eine breite Spanne von Tools für die Erkennung spezieller Fehler. Dafür die Diagnose häufig ein Vergleich der Ergebnisse aus verschiedenen Messungen notwendig ist, ermöglicht das System die Gegenüberstellung bestimmter Messwerte beispielsweise für Vibration, Ölanalyse, Druckänderung und Temperatur. Erweiterte Analyse-Tools zeigen dem Anwender automatisch an, welche Maschinenkomponente ein Problem hervorgerufen hat.[2]

2 Überwachungstechniken

2.1 Schwingungsanalyse

Maschinen-Vibrationen ist einer der besten Indikatoren für die gesamte Maschine für die meisten Arten von rotierenden Maschinen. Wir verwenden Vibrati-

onsanalyse fast jeden Tag in unserem Leben. Wir klopfen einer Wassermelone, um die Reife aus den Geräuschen und Vibrationen der Melone zu beurteilen. Wie ist es mit einer rotierenden Maschine zum Beispiel ein Haushalt Mixer oder ein Auto-Motor? Diese Maschinen produzieren eine Art von Lärm, dass im Zusammenhang mit der Drehzahl steht, z. B. die Umsetzung des Mixers in einen schnelleren Modus produziert Lärm auf einer höheren Frequenz. Neue Klänge oder eine Veränderung in der Frequenz (Tonhöhe / Zyklen pro Sekunde Ton) von Geräuschen kann ein Problem bedeuten.

Bei der Vibrationsanalyse werden die Schwingungen von drehenden Teilen (z.B. Achsen) aufgezeichnet und analysiert. Anhand der Schwingmuster sowie der Schwingungsamplituden lassen sich Rückschlüsse auf den Zustand der drehenden Teile sowie der Lager ziehen. Hierdurch lassen sich notwendige Reparaturen im Vorfeld erkennen und vor Ausfall der Maschine durchführen. Dadurch können Stillstandszeiten reduziert bzw. in produktionsfreie Zeiten verlagert werden.

Die gefilterte Vibration, als spektrale Komponente, ist ein ausgezeichneter Indikator für Schwingstärke und Zustand der Maschine. Ein Vibrationsspektrum, graph Amplitude der Schwingung der Frequenz, zeigt die spezifische Maschinenfehler sowie falsche Ausrichtung, Ungleichgewicht, und Lagerverschleiß. Das grundlegende Prinzip von PDM besteht aus regelmäßigen Messungen der Vibrationen mit ständigen oder tragbaren Instrumenten.[3]

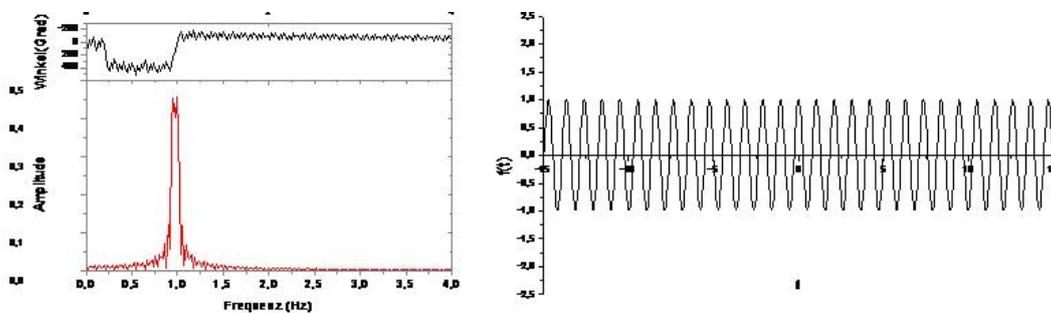


Abbildung 2. a) Vibrationsspektrum b) Zum Vergleich. Analyse einer einzigen Sinus Kurve.

Analyse der einzelnen Frequenz Ebenen können zum Fehlerdiagnose verwendet werden, da innerhalb der Maschine verschiedene Komponenten in verschiedenen Frequenzen vibriert. Die Frequenzen unterscheiden sich auch in gut und schlecht Zustand. Abbildung 3. zeigt so eine Vibrationspectrum.

2.2 Ultraschallmessungen

Ultraschall-Integritätsprüfung ist eine Methode zur Bestimmung der Homogenität des Materials von starren Körpern. Eine Standardanwendung dieser Messme-

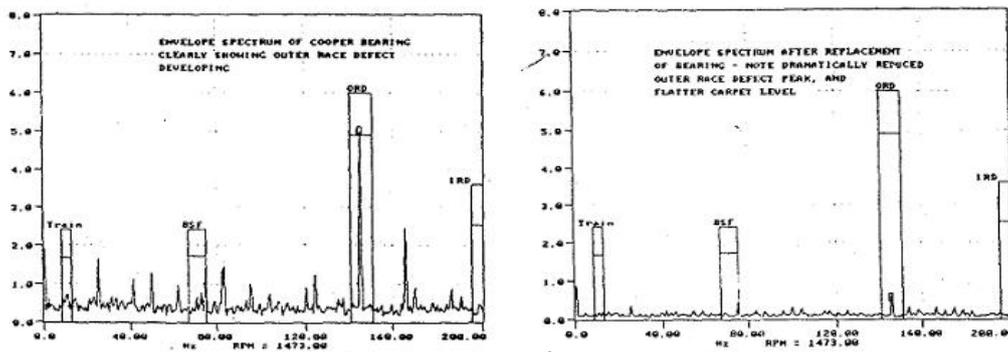


Abbildung 3. schlechter Zustand(links) guter Zustand(rechts)

thode ist das Testen von Ortbetonbohrpfählen. Abweichungen in der Homogenität des Materials werden durch Abweichungen der Ausbreitungsgeschwindigkeit von Ultraschallwellen festgestellt. Diese Abweichungen werden als Anomalien bezeichnet. Normen (z.B. USA-ASTM D6760-02 oder France-NF P94-160-1) legen die technischen Voraussetzungen zur Durchführung dieser Tests sowie die Eigenschaften der Messgeräte fest. Am Beispiel(Bild 1) soll einer Airbagabdeckung gezeigt werden, daß man mit Luftutraschall verdeckte Strukturen (weiße Ellipsen) in Bauteilen sichtbar machen kann.

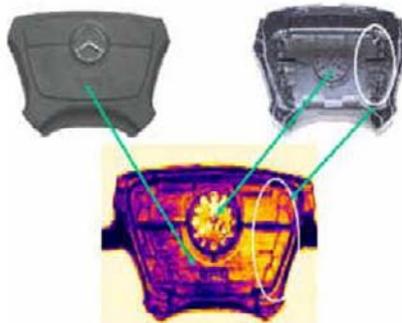


Abbildung 4. Strukturuntersuchung an einer Airbagabdeckung

2.3 Ölanalyse

Diese Art der Analyse, für die Öl als ein Informationsträger dient, ermöglicht es, den Zustand der Geräte zu überprüfen. Die Folgemaßnahmen bestehen aus einer Reihe standardisierten mechanisch-chemischer Analysen, die auf Basis des

Schmierstoffes und seiner Betriebsbedingungen ausgewählt werden. Diese Analysen können parallel zu Vibrationsanalyse-Messungen durchgeführt werden, und ermöglichen es, dass kritische Komponenten mit einem präventiven Wartungsprogramm behandelt werden.

2.4 Infrared Thermography

Thermografie ist die Sichtbarmachung der von einem Objekt abgestrahlten thermischen Energie mit Hilfe einer Kamera. Wärmebildkameras arbeiten im Infrarotbereich. Die Strahlen sind daher für das menschliche Auge nicht sichtbar. Infrarotkameras erzeugen Bilder der unsichtbaren Infrarot- bzw. Wärmestrahlung und ermöglichen damit eine präzise thermische Abbildung der Wärmeverteilung, resp. -abstrahlung des gemessenen Objektes. Mittels Infrarot -Thermografie können Wärmeentwicklungen in Maschinen sichtbar gemacht werden. Die Thermographieaufnahmen werden normalerweise zusammen mit den Schwingungsmessungen gemacht, so dass eine zusammenhängende Analyse erstellt werden kann.

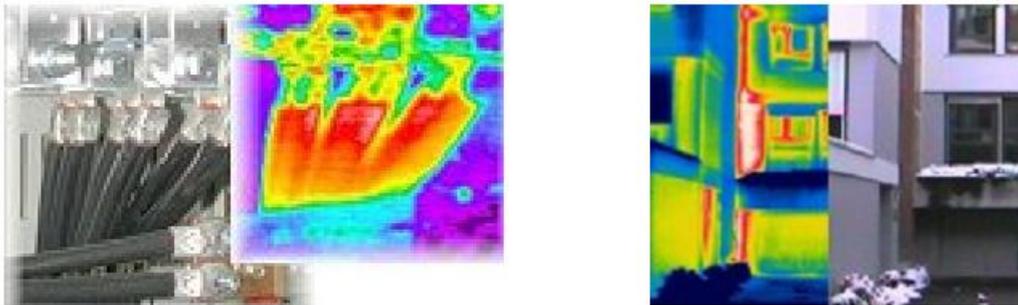


Abbildung 5. Infrared Analysen durch Elektrische Anlagen(links) Bauten(rechts)

3 Beschleunigungssensor (Accelerometer)

Ein Beschleunigungssensor ist ein Sensor, der die Beschleunigung misst, indem die auf eine Testmasse wirkende Trägheitskraft bestimmt wird. Somit kann z. B. bestimmt werden, ob eine Geschwindigkeitszunahme oder -abnahme stattfindet. Sie sind Sensoren für die Gesundheitsüberwachung von rotierenden Maschinen wie Pumpen, Fans, Rollen, Kompressoren und Kühltürme etc.

Für die meisten Anwendungen im Bereich der Maschinenüberwachung und Fehlerdiagnose haben sich Beschleunigungsaufnehmer nach dem piezoelektrischen

Prinzip durchgesetzt. Der sehr große Dynamikbereich und der breite nutzbare Frequenzbereich sind dafür ausschlaggebend. Beschleunigungsaufnehmer werden von einer Vielzahl von Firmen angeboten und unterscheiden sich je nach benötigtem Frequenzbereich, Dynamikbereich, Größe, Befestigung usw. Der große Nachteil piezoelektrischer Beschleunigungsaufnehmer ist ihr hochohmiger Ausgang, der insbesondere bei großen Leitungslängen zu einer hohen Störempfindlichkeit führt. Da die Beschleunigung proportional einer elektrischen Ladung ist, wird ein spezieller Vorverstärker, ein sogenannte Ladungsverstärker, benötigt. Dank heutiger Halbleitertechnologie bietet die Industrie Sensoren mit im Sensorgehäuse integriertem Ladungsverstärker, so dass dieser Nachteil für viele Anwendungen nicht mehr relevant ist.[4]



Abbildung 6. Beschleunigungssensor

4 Schwingungsmessung

Jeder weiß, wie eine Porzellantasse klingt, wenn sie einen Riss hat. Den Unterschied zur unbeschädigten hört man sofort. Hierbei wird die Schwingung durch Anschlagen erzeugt, vom Ohr aufgenommen und mit dem im Gehirn durch Erfahrung abgespeicherten Klangmuster der intakten Tasse verglichen und ausgewertet.

Die allgemein Aufgabe der Schwingungsmessung ist das erfassen von Frequenzen und diese zu interpretieren. Verschiedene Methoden sind vergleichbar um die Eigenschaften von erfassenden Frequenzen zu interpretieren. Die folgenden Eigenschaften sind zu vergleichen:

4.1 Leistungsspektrum

Die Aufspaltung eines Signals in seine Frequenzanteile (Frequenzanalyse) liefert sein Spektrum. Da die einzelnen Frequenzanteile untereinander verschiedene Phasenlagen haben können, ist ein Spektrum im Allgemeinen komplex, d. h. es kann in ein Betragsspektrum und ein Phasenspektrum zerlegt werden. Energie- und Leistungsspektren sind hingegen rein reell und repräsentieren die Frequenzverteilung der Energie bzw. Leistung. In der Praxis werden Spektren entweder durch Filterbänke (Serie von Bandpassfiltern, meist Oktav- oder Terzfilter) oder mit Hilfe der diskreten Fourier-Transformation (DFT) gebildet. Ihre Auflösung ist daher begrenzt, so dass die diskreten Werte eines Spektrums jeweils den Inhalt eines endlich breiten Frequenzbandes inklusive Überlappungen repräsentieren (wesentlich bei kontinuierlichen Spektren, wie z. B. Rauschen). Die spektrale Leistungsdichte (power spectral density, PSD) gibt hingegen die Leistung je Einheits-Frequenzband an, die unabhängig von der Auflösung ist.[5] Standard-Leistungsspektrum benutzt Welch's method. Data ist normiert auf den Mittelwert vor Spektrum Schätzung und Merkmalsvektoren (bestehend aus spektralen Amplituden) sind normiert auf w.r.t Mittelwert und Standardabweichung (damit nur Sensibilität für das Spektrum Form).[3]

4.2 Hüllkurvenanalyse

Die Hüllkurvenanalyse ist eine bewährte Methode, um periodische Stöße im Schwingungssignal einer Maschine zu erkennen. Auch periodische Stöße mit geringstem Energieinhalt, die darüber hinaus durch das Gesamt-Schwingungssignal überdeckt werden, können mit ihr detektiert werden. Die Hüllkurvenanalyse ist daher zu einem der wichtigsten Werkzeuge von Instandhaltungsfachleuten geworden.

Wälzlagerschäden sind ein häufiges Problem an industriellen Maschinen. Gerade sie lassen sich ideal mit der Hüllkurvenanalyse detektieren. Entsteht ein Schaden im Wälzlager, findet eine Amplitudenmodulation des vom Wälzlager emittierten Schwingungssignals aufgrund sich periodisch ändernder Kräfte statt. Bei der Hüllkurvenanalyse werden die niederfrequenten Maschinen-schwingungen inklusive der Lagerschadensfrequenzen ausgefiltert, um die periodischen (weitaus schwächeren) Hüllkurvensignale zu extrahieren. Das Spektrum des resultierenden Signals erlaubt im Vergleich mit anderen Analysemethoden ein deutlich früheres Erkennen sich anbahnender Schäden.[3]

Eine Messung der Zeit-Serie wurde mit der Hilbert-Transformation demoduliert und vom HT gereinigten Signal (angeblich mit Informationen über das regelmäßige impulsive Verhalten) wurde ein Spektrum festgestellt. Vor der Demodulation wurde Bandpass-Filter im Intervall 125 - 250 Hz (mit einem Wavelet Zerlegung mit Daubechies-Wavelets der Ordnung 4) durchgeführt: Getriebe

Maschenöffnung Frequenzen werden in diesem Band anwesend und Impulse werden durch Lochfraß als sidebands erwartet. Zum Vergleich wurde diese Vorfilter Schritt in einem anderen Datensatz ausgelassen.[3]

4.3 Autoregressiver Modellierung

Eine weitere Möglichkeit zur Nutzung zweiter Ordnung Korrelation als eine Funktion ist die Modellierung der Zeitreihe mit einer Autoregressive-Modell (AR-Modell). Für den Vergleich mit anderen Funktionen wurde AR (64)-Modell verwendet um alle Informationen zu extrahieren und Modell-Koeffizienten wurden als Funktionen verwendet.[3]

Beim autoregressiven Modell wird die Zeitreihe des abgeleiteten Schwingungssignals als Ausgang eines linearen, optimalen Filters modelliert, wobei das Eingangssignal weisses Rauschen darstellt. Aufgrund dieses Modells werden die Koeffizienten dieses Filters so bestimmt, dass für jeden aktuellen Signalwert eine optimale Schätzung mit Hilfe von Signalwerten zu früheren Zeitpunkten erfolgt; d.h. die beobachtete Zeitreihe lässt sich durch ihre eigene Vergangenheit ausdrücken. Die Filterkoeffizienten sind gleichzeitig die sog. autoregressiven Koeffizienten. Dieses Verfahren, das zur Gruppe der parametrischen Modelle gehört, führt zu einer maximalen Datenreduktion bei minimalem Informationsverlust.

Zur Spektralschätzung wird das AR-Modell in den meisten Fällen verwendet. Erstens ist zur Bestimmung der Koeffizienten eines AR-Modells ein lineares Gleichungssystem zu lösen. Zweitens stützt sich das AR-Modell auf ein all-pole-System ab und deswegen existieren für das durch das AR-Modell berechnete Leistungsdichtespektrum im Gegensatz zu den traditionellen Schätzungsverfahren keine Beschränkungen bezüglich des Auflösungsvermögens im Frequenzbereich. AR-Modelle eignen sich besonders dazu, scharfe Spektrallinien periodischer Signale durch Polstellen des Nennerpolynoms zu approximieren. Sie sind demnach gut zur Schätzung von Spektren mit ausgeprägten diskreten Spektrallinien geeignet.[4]

4.4 MUSIK Spektrum Schätzung

Wenn eine Zeitreihe als ein Modell der Sinusoide plus Lärm modelliert wird, kann eine MUSIC (Multiple Signal Klassifikation) Frequenz Schätzers verwendet werden um wichtige spektrale Komponenten zu konzentrieren. MUSIC Funktionen ermöglichen gute Trennbarkeit, während die Funktion Größe relativ gering ist. Die Amplituden werden eine Diskriminante Indikator bei einer endlichen Anzahl von diskreten Frequenzen.[6]

5 Vorverarbeitung

Die durch Sensoren gewonnene Messwerte sind noch nicht kalibriert und enthalten mitunter noch zahlreiche Datenfehler, die die Analyse unnötig stören. Zudem arbeiten die Analyseprogramme meist mit einer geringeren Abstrakte als die Datenerfassung. Vor der Analyse der Daten ist also noch eine Kalibrierung und Bereinigung der Daten nötig.[7]

- Schritte der Vorverarbeitung (Auswahl):
 - A) problemspezifische Transformationen
 - B) lineare Transformationen
 - C) Berücksichtigung von Invarianzen
 - D) Dimensionsreduktion
 - E) Beseitigung von Mängeln in Daten

PCA(Principal Component Analysis) PCA ist eine weit verbreitete und vielfältig anwendbare Methode der Dimensionsreduktion, und der Merkmalsextraktion. Die Merkmalsextraktion und Datenkompression stellt einen wichtigen Zwischenschritt bei der Verarbeitung von Daten in ubiquitären Systemen dar. Sie wird benutzt zur Komprimierung, zum Entrauschen von Daten oder allgemein als Vorverarbeitung bei Klassifikations-, Regressions- oder Quellentrennungsaufgaben.

PCA richtet die Achsen entlang der größten Varianz aus. Dazu wird von den Daten die Kovarianz-Matrix berechnet und aus dieser wiederum die Eigenvektoren mit zugehörigen λ -Werten. Ordnet man nun die Eigenvektoren absteigend nach der Größe der Eigenwerte erhält man die Komponenten. Hier kann man Dimensionen reduzieren bzw. das Kaiser-Goodman-Kriterium (nur Eigenvektoren mit λ -Werten > 1) anwenden. Die neuen Achsen sind nun die Eigenvektoren.[8]

6 Klassifizierung

Bevor Ausgangsdaten verarbeitet werden, sollten sie klassifiziert oder gruppiert werden, da es nicht zweckmäßig ist, jeder Dateneintrag zu interpretieren. In diesem Zusammenhang ist die Umsetzung der Künstlichen Intelligenz auf Ressourcen begrenzten Plattformen ein interessantes Forschungsgebiet. Eine der statischen Methoden, die angewendet werden können, ist das Support Vector Machines (SVMs).

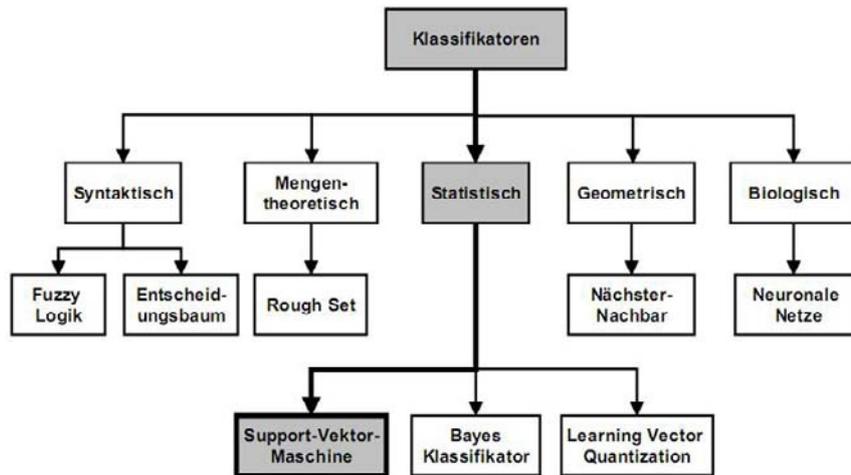
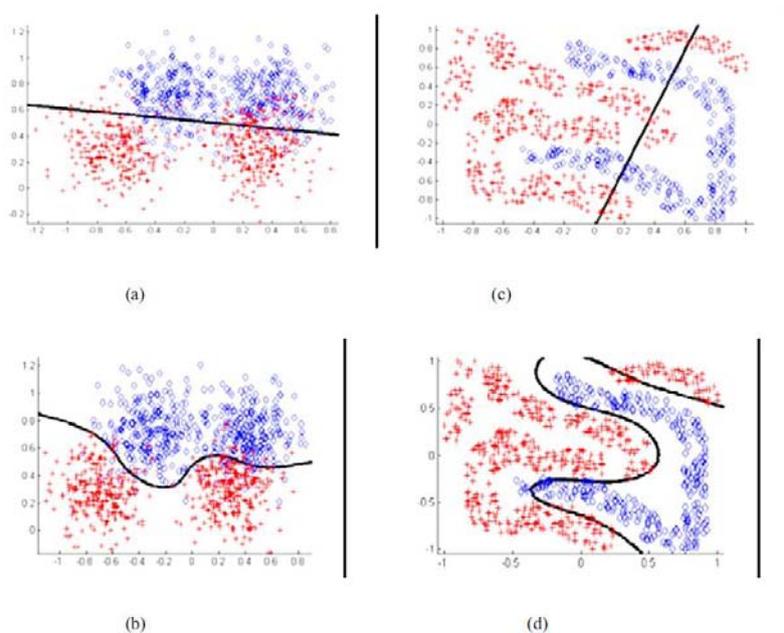


Abbildung 7. Übersicht der verschiedenen Klassifikatoren

Support Vector Data Description (SVM) Support Vector Machine (SVM) "Stützvektormaschine" ist ein Lernalgorithmus zur Klassifizierung von Objekten. Eine Support Vector Machine (SVM) ist in der Informatik ein Klassifikator, also ein Programmobjekt oder eine Summe von Klassen. Die Grundidee der Support Vector Machine (SVM) Stützvektormaschine ist, die Hyperebene "maximum Margin hyperplan" zu finden, die die Daten der Klassen oder Objekte



trennt.

Abbildung 8. Klassifikation synthetischer Daten mit linearen und nichtlinearen SVMs.

Einen SVM-Algorithmus für lineare Klassifikation kann man für nicht-lineare Klassifikation benutzen, indem man das Skalarprodukt durch eine geeignete Kern-Funktion (Gaussian Kernel) K ersetzt, und somit implizit in einem anderen (möglicherweise dimensionaleren) Raum rechnet. Dies wird häufig als der Kernel Trick bezeichnet. [7]

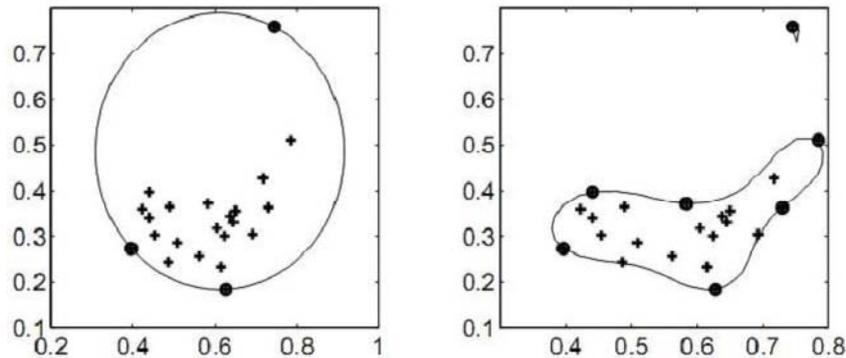


Abbildung 9. a) Beschreibung der Daten ohne Gaussian Kernel b) mit Gaussian Kernel

Hier sind SVMs aufgrund ihrer Robustheit und guten Sparsity-Eigenschaften verwendet. SVMs sind allerdings auf energiesparenden 8-bit MCU schwer oder gar nicht zu realisieren. RSVMs sind einfacher, energiesparender und kostengünstiger. RSVMs wurden vor Kurzem als eine Methode zum maschinellen Lernen vorgeschlagen. Sie sind mit weniger Eingängen implementiert. Es hat sich gezeigt, dass die Zeit zur Berechnung und die Speichernutzung viel weniger durch RSVM als für diejenigen, die durch SVM, die einen Standard-Kernel verwenden [9].

Literatur

1. Wikipedia.: Instandhaltung. <http://de.wikipedia.org/wiki/Instandhaltung> 10. Januar 2009
2. Piotrowski, J.: Pro-Active Maintenance for Pumps (2001), Chapter 5 Types of Maintenance Programs. <http://www.pump-zone.com>. Reprinted with permission of Pump-Zone.com
3. David Tax, Alexander Ypma, R.P.W.D.: 4- Support Vector Data Description Applied to Machine Vibration Analysis (1999). <http://citeseer.ist.psu.edu/article/tax99support.html>
4. Liu, M.E.Z.: Signalverarbeitungsverfahren und virtuelle Instrumente zur Messung von elektrischen Signalen und zur Fehlerdiagnose an Maschine (2006). http://deposit.ddb.de/cgi-bin/dokserv?idn=978193873&dok_var=d1&dok_ext=pdf&filename=978193873.pdf

5. Kjaer, B.: Frequency Analysis. Rev. Sept. (1987). <http://www.motorlexikon.de/?I=6178&R=S>
6. Manolakis., J.P.D.: Digital Signal Processing- principles, algorithmis and applications.(1992). 2nd ed. MacMillan Publ.
7. Ölaf Dierks: Optimierung der Gezeitanalyse mit Daten eines Dual Sphere Supraleitgravimeters (2000). http://home.graffiti.net/oлие_d:graffiti.net/Diplomarbeit_Olaf_Dierks.pdf
8. Forsyth, D., Ponce., J.: Computer Vision. A Modern Approach. Prentice Hall PTR, 2002
9. Aliaksei Kerhet, Mingqing Hu, F.L.A.B.D.P.: Svm-like algorithms and architectures for embedded computational intelligence technical report, march 2007. <http://eprints.biblio.unitn.it/archive/00001174/01/012.pdf>
10. Tübingen, U.: Analyse nach harmonischen schwingungen 2008. http://www.uni-tuebingen.de/uni/pki/skript_ppt_08/Schwingungen_FFT_mech_Modell_08.ppt
11. und Kjaer Vibro, B.: Hüllkurvenanalyse (2009). <http://www.bkvibro.de/Main/L%C3%B6sungen+%26+Anwendungen/Messtechniken/H%C3%BC11kurvenanalyse>

Verarbeitung und Erzeugung von ungewissen und unscharfen Informationen in ubiquitären Systemen

Nadia Ahmed

Telecooperation Office, Institut für Telematik, Universität Karlsruhe (TH)
Betreuer: Martin Berchtold

1 Einleitung

In Ubiquitous Computing spielen Informationen über die reale Welt eine grosse Rolle. Durch Sensoren werden Beobachtungen über das Benutzerverhalten oder die Umwelt gesammelt um sie für Schlussfolgerungen zu verwenden. Zum Beispiel könnte ein Handy aus dem Wissen dass es sich in einem Besprechungsraum befindet und es in diesem Raum laut ist, darauf schliessen dass sich der Benutzer in einem Meeting befindet und nicht gestört werden will.

Für solche Schlussfolgerungen ist formale Logik geeignet. Sie ermöglicht die Modellierung komplexer Zusammenhänge und das automatisierte Schliessen aus dem formalisierten Wissen. Die klassische Logik bestehend aus Regeln für Syntax, Semantik und Schlussmechanismen (Kalküle). Beispiele hierfür sind Aussagenlogik und Prädikatenlogiken (erster oder höherer Stufe).

Das Problem dabei ist, dass die Informationen unvollständig, ungenau, mangelhaft, unsicher oder widersprüchlich sein können. Es ist schwer in Beobachtungen über die reale Welt solche Ungewissheiten zu vermeiden. Zum Beispiel kann mit Hilfe eines Mikrofons die Lautstärke in einem Raum bestimmt werden. Der Messwert kann allerdings ungenau oder durch Störeinflüsse verfälscht sein. Zudem können nur binäre Informationen wie "Ist es hier laut? Ja/Nein" verarbeitet werden, jedoch möchte man vielleicht fließende Übergänge in den Entscheidungen verwenden.

Ein weiteres Problem besteht darin dass man in der Syntax der klassischen Logiken nicht jeden Aspekt des Denkens modellieren kann. Zum Beispiel könnte man ausdrücken wollen dass wenn eine Person Student ist, sie vermutlich auch jung ist. Mit der Formel $\forall x \text{ student}(x) \rightarrow \text{jung}(x)$ in Prädikatenlogik erster Stufe kann man diese Vagheit allerdings nicht berücksichtigen. Die Formel würde aussagen dass jeder Student jung ist, was im Allgemeinen aber falsch ist.

Ein mögliche Lösung besteht darin, bei der Schlussfolgerung Methoden aus der Wahrscheinlichkeitstheorie oder Unscharfen Logik anzuwenden. Dieser Text behandelt genau diesen Lösungsansatz. Das folgende Kapitel bietet eine Einführung in einige nichtklassische Logiken, die geeignet sind um diese Probleme zu lösen.

2 Logiken

2.1 Modallogik

Clarence Irving Lewis hat im Jahr 1918 ein formales axiomatisches modallogisches System vorgestellt. Seit dieser ersten Version hat sich die Modallogik weiterentwickelt und unter anderem haben Kurt Gödel und Saul Kripke wichtige Beiträge zur Entwicklung geleistet.

In der Modallogik hat man die Möglichkeit auszudrücken dass etwas „möglicherweise“ bzw. „notwendigerweise“ wahr oder falsch ist. Also könnte man zum Beispiel eine Aussage wie „Eine Tasse ist möglicherweise mit Kaffee gefüllt“ formalisieren. Ausführlichere Darstellung der modalen Logik findet man in [1]. Die folgende Darstellung orientiert sich u.a. an dem Skriptum [2].

Syntax. Um solche Aussagen zu formalisieren werden zwei zusätzliche Symbole, \Box (gesprochen Box) und \Diamond (gesprochen Diamond), verwendet. Dabei bedeutet $\Diamond p$ dass eine Aussage p möglicherweise wahr ist und $\Box p$ dass die Aussage p notwendigerweise wahr ist. Das formalisierte Beispiel lautet dann $\forall x \text{ tasse}(x) \rightarrow \Diamond \text{voll}(x, \text{kaffee})$.

Semantik. Dabei ist $\Diamond \text{voll}(x, \text{kaffee})$ erfüllt wenn es für x eine "denkbare Situation" gibt in der x voll mit Kaffee ist. Was nichts anderes als "x ist mit Kaffee befüllbar" bedeutet. Wenn also x ein geeignetes Gefäß ist wird die Aussage wahr.

Die "denkbaren Situationen" werden formal als erreichbare Welten oder Zustände bezeichnet. Was das genau bedeutet wird nun weiter erläutert.

Um einer Formel eine Semantik zuzuweisen, werden Kripke-Strukturen verwendet. Eine Kripke-Struktur besteht aus einer Menge von Welten (oder Zuständen) und einer Übergangsrelation zwischen den Welten. Eine Welt wiederum ist eine prädikatenlogische Struktur, wobei alle Welten sich das selbe Universum (manchmal auch Domäne genannt) teilen.

Hat man beispielsweise ein Universum das aus Personen besteht und ein einstelliges Prädikat *busy* das angeben soll ob eine Person beschäftigt ist, so kann für die selbe Person x in einer Welt $\text{busy}(x)$ wahr sein und in einer anderen Welt falsch. Als Welten könnte man nun Zeitpunkte und als Übergangsrelation die temporale Ordnung festlegen. D. h. es existiert ein Übergang zwischen W_1 und W_2 genau dann, wenn W_1 früher ist als W_2 .

Eine Formel $\Box p$ wird wahr wenn die Formel p in allen erreichbaren Welten wahr wird, mit der Struktur im Beispiel würde das bedeuten dass p zu jedem zukünftigen Zeitpunkt gelten muss.

Eine Formel $\Diamond p$ wird wahr wenn es eine erreichbare Welt gibt, in der p gilt. Im Beispiel würde das bedeuten, dass es einen zukünftigen Zeitpunkt gibt indem p gilt.

Dabei kann man sich vorstellen dass man sich in einer Welt befindet (der Ausgangspunkt) und in dieser Welt die Formel auf Richtigkeit prüft. Dabei wird eine Formel allgemein wahr bezüglich einer Kripke-Struktur betrachtet wenn diese in jeder Welt wahr ist (also für einen beliebigen Ausgangspunkt).

Die Formel $\forall x (sick(x) \rightarrow \diamond \neg sick(x))$ würde dann mit der obigen Kripke-Struktur bedeuten, dass wenn eine Person krank ist, es einen zukünftigen Zeitpunkt gibt an dem sie nichtmehr krank ist.

Fazit. Man kann Schlussfolgerungen mit Modallogik verwenden um Modalitäten beim formalisieren berücksichtigen zu können. Wenn man verschiedene Situationen betrachten will, in denen unterschiedliche Fakten gelten und die Relation zwischen diesen Situationen mit berücksichtigen will, so bietet die Modallogik die richtigen Hilfsmittel.

Um die Probleme der Ungewissheit und Ungenauigkeit beim Ubiquitous Computing zu lösen ist die Modallogik jedoch nur sehr begrenzt geeignet. Allerdings bietet eine Kombination mit anderen Ansätzen wie Unscharfe Logik (Fuzzy Logik) oder Probabilistische Logik, die im Anschluss vorgestellt werden, grosses Potenzial. Diese Kombination kann man durch Syntaxerweiterungen der jeweiligen Logiken erreichen.

2.2 Unscharfe Logik (Fuzzy Logic)

Die Unscharfe Logik ermöglicht das Modellieren von Ungewissheit in der Form von Unschärfe. Das heisst eine Formel kann nicht einfach nur wahr oder falsch sein, sondern auch halb wahr oder ziemlich wahr.

Ausführliche Informationen über dieses Thema findet man in [3].

Die Theorie über unscharfe Mengen und damit der Grundstein für die Unscharfe Logik wurde 1965 von Lotfi Asker Zadeh entwickelt. Obwohl die Idee dass zwischen wahr und falsch noch ein dritter Bereich ist schon von dem antiken Philosoph Platon postuliert wurde.

Die Syntax der Unscharfen Logik ist identisch mit der Prädikatenlogik, lediglich in der Semantik unterscheiden sich die Logiken. Normalerweise wird ein einstelliges Prädikat als Teilmenge des Universums interpretiert, in der Unscharfen Logik wird es als unscharfe Menge interpretiert. Was genau eine unscharfe Menge ist wird nun weiter erläutert.

Zugehörigkeitsfunktionen. Zu einem Universum U heisst eine Abbildung $\mu : U \rightarrow [0, 1]$ Zugehörigkeitsfunktion. Die Absicht dahinter ist jedem Element aus dem Universum einen Zugehörigkeitsgrad zwischen 0 und 1 zuzuweisen.

Hat man beispielsweise ein Universum, dass aus Personen besteht, so könnte man eine Zugehörigkeitsfunktion definieren die angibt wie alt eine Person ist. Der Wert 0.5 würde besagen dass die Person mittelalt ist und könnte ein Alter von etwa 40 Jahren bedeuten.

Unscharfe Mengen. Eine unscharfe Menge \tilde{A} wird durch eine Zugehörigkeitsfunktion μ festgelegt. Formal wird eine unscharfe Menge \tilde{A} definiert durch $\tilde{A} := \{(x, \mu(x)) : x \in U\}$, was der Darstellung von μ als Relation entspricht.

Ist beispielsweise $U = \{Patrick, Steve, Jane\}$ so könnte:

$$\tilde{A} = \{(Patrick, 0.3), (Steve, 0.35), (Jane, 0.6)\}$$

einer unscharfen Menge für das Alter der Personen aus U entsprechen. In der Logik kann man nun ein Prädikat alt einsetzen, wobei $alt(x)$ nicht wahr oder falsch ist, sondern eine Zahl zwischen 0 und 1. Dabei steht 0 für falsch und 1 für wahr, dazwischen gibt es noch unendlich viele Abstufungen.

Logische Operationen. Es stellt sich die Frage wie man nun mit logischen Operatoren wie \wedge oder \neg umgehen soll, wenn sie nichtmehr auf Wahrheitswerte sondern auf reelle Zahlen zwischen 0 und 1 angewendet werden. Hierfür gibt es eine grosse Vielfalt an Möglichkeiten, die in der weiterführenden Literatur (häufig als s- bzw. t-Normen bezeichnet) ausführlich behandelt werden. Hier werde ich nur einen intuitiven Ansatz vorstellen.

Bei einer \wedge -Verknüpfung kann man das Minimum beider Operanten nehmen. Das hat zur Folge dass sobald ein Operant komplett falsch wird auch das Ergebnis komplett falsch wird. Dies entspricht der Intuition aus der klassischen Logik. Da auch der Allquantor intuitiv einer Konjunktion entspricht, kann man hier genauso vorgehen. Das Ergebnis der Formel $\forall x p(x)$ ist dann das Minimum aller $\mu_p(x)$ für $x \in U$, wenn das Prädikat p mit der Zugehörigkeitsfunktion μ_p interpretiert wird.

Analog kann man für die Disjunktion und den Existenzquantor das Maximum verwenden.

Für die Negation ist es naheliegend für den Zugehörigkeitsgrad $\mu(x)$ einfach $1 - \mu(x)$ zu verwenden. Somit wird wahr zu falsch und umgekehrt und die Regeln aus der klassischen Logik bleiben erhalten.

Anwendung: Verteilte Aktivitätserkennung. Eine Kooperation der University of Twente in den Niederlanden und der ETH Zürich in der Schweiz hat ein Projekt hervorgebracht das die Effizienz in Produktionsanlagen verbessern soll. Dabei wird auch Fuzzy Logic eingesetzt. Für weitere Informationen siehe [4].

Ein drahtloses Sensornetz (WSN: Wireless Sensor Network) wird in einer Automobilproduktionsanlage eingesetzt, um Aktivitäten, die für den Zusammenbau von Bedeutung sind, zu erkennen. Dieses Sensornetz bildet ein verteiltes System. Drahtlose Sensor-Knoten in diesem Netz sind beispielsweise an der Kleidung oder den Werkzeugen der Mitarbeiter angebracht. Das System soll erkennen welche Aktivitäten die Arbeiter gerade durchführen und ihnen dann situationsbedingt assistieren. Um dieses Ziel zu erreichen werden die Sensordaten der einzelnen Knoten mittels unscharfer Logik verarbeitet und kombiniert.

Abbildung 2 zeigt schematisch den Ablauf der Auswertung der Sensordaten und das schließen auf die Aktivitäten. Dabei ist $D = \{D_1, \dots, D_d\}$ die Menge der Sensoren. Jeder Sensor D_i meldet ein Menge $E_i = \{E_{i1}, \dots, E_{ie_i}\}$ von Ereignissen. Die Ereignisse der verschiedenen Sensoren werden dann kombiniert zu



Abb. 1. Sensoren angebaut bei dem Mitarbeiter, bei seiner Werkzeuge und bei dem Auto: Drahtlose Sensor-Knoten

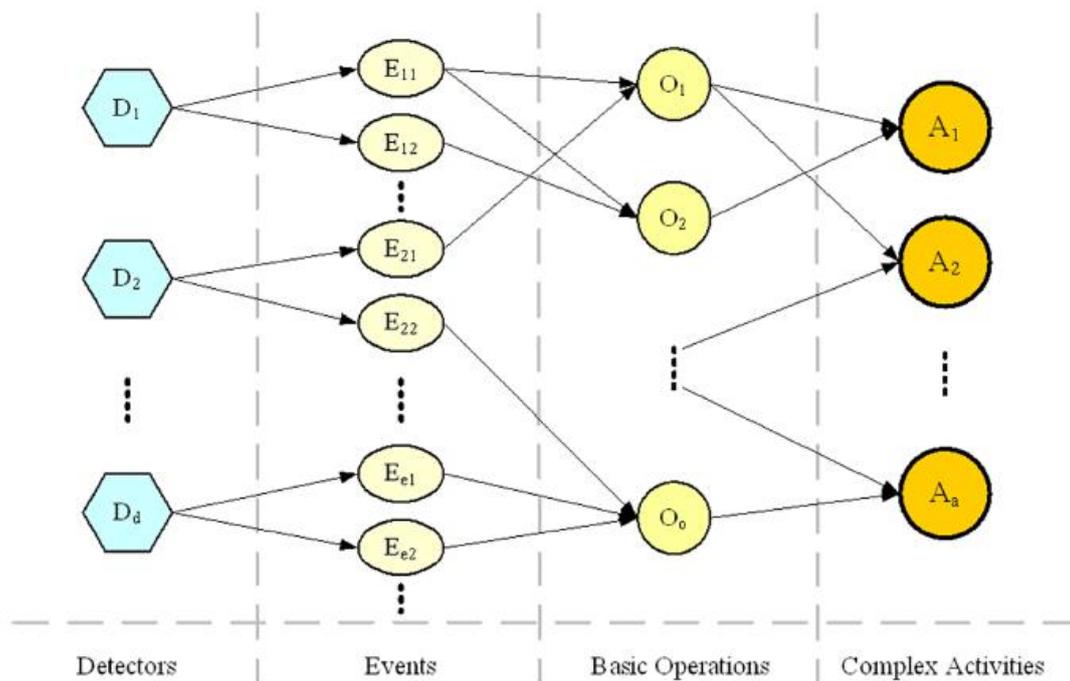


Abb. 2. Erkennung von Verteilte Aktivitäten mit Fuzzy WSN

Operationen $O = \{O_1, \dots, O_o\}$. Diese Identifizierung der Operationen ist der erste Schritt in der Datenfusion. Die zweite Stufe abstrahiert von diesen einfachen Operationen zu komplexeren Aktivitäten $A = \{A_1, \dots, A_a\}$.

Bei diesem Prozess gibt es zwei Probleme, bei denen unscharfe Logik helfen kann. Das erste Problem liegt in der Ungenauigkeit der Sensoren, so kann es passieren, dass nicht eintretende Ereignisse fälschlicherweise detektiert oder eintretende Ereignisse nicht detektiert werden. Um dieses Problem zu verringern kann man die Ereignismenge als unscharfe Menge ansehen. Das heißt der Sensor sagt wie sehr ein bestimmtes Ereignis eintritt. Dies liefert eine Zuge-

hörigkeitsfunktion die jedem möglichen Ereigniss einen Wert zwischen 0 und 1 liefert.

Das zweite Problem tritt während der Klassifikation von Operationen und Aktivitäten auf. So kann es passieren dass der Klassifikationsprozess nicht eindeutig entscheiden kann welche Aktivität vorliegt, da die aktuellen Operationen mehrere Schlüsse zulassen. Hier kann mit Hilfe der unscharfen Logik bestimmt werden wie sehr welche Aktivität momentan vorliegt. Um die Informationen der Sensoren zu Operationen zu fusionieren oder aus den Operationen auf die Aktivitäten zu schließen kann man nun die logischen Operatoren der unscharfen Logik verwenden.

Fazit. Die Formalisierungsmöglichkeiten der Unscharfen Logik eignen sich um Sensordaten in die Schlussfolgerungen mit einfließen zu lassen. Beispielsweise könnte man die Temperatur, Lautstärke oder Helligkeit die Sensoren messen als unscharfe Mengen verwenden, indem man die Messwerte in den Bereich zwischen 0 und 1 skaliert.

Der Bereich der Unscharfen Logik hat sich seit mehr als vierzig Jahren, durch viele erfolgreiche Anwendungen in verschiedenen Gebieten wie Automobilindustrie, künstliche Intelligenz, Medizin und Verhaltenswissenschaft entwickelt.

Das Problem der Vagheit kann somit durch den Einsatz von Fuzzy Logic beseitigt werden, jedoch kann man noch keine Aussagen über Unsicherheiten mit Wahrscheinlichkeiten ausdrücken. Hierfür sind die Possibilistische bzw. Probabilistische Logik geeignet.

2.3 Possibilistische Logik

Einer Formel p der Prädikatenlogik erster Stufe oder der Aussagenlogik werden zwei Werte zwischen 0 und 1 zugewiesen, die die Ungewissheit ausdrücken sollen. Dies sind der Possibilitätsgrad $\Pi(p)$ und der Notwendigkeitsgrad $N(p)$.

Ist $N(p) = 1$, so muss die Formel p auf jeden Fall wahr sein. Die Notwendigkeit ist in dem Fall maximal. Ist $\Pi(p) = 0$, so ist es unmöglich dass die Formel p wahr ist. Die Possibilität ist minimal.

Um die beiden Werte zu verknüpfen, stellt man für eine beliebige Formel p die Forderung $\Pi(p) = 1 - N(\neg p)$. Daraus folgt insbesondere $\Pi(p) = 0 \Rightarrow N(\neg p) = 1$, was bedeutet, dass wenn eine Formel unmöglich wahr sein kann, die negierte Formel notwendigerweise wahr sein muss.

Gilt für eine Formel p dass die Possibilität von p und von $\neg p$ beide 1 sind ($\Pi(p) = \Pi(\neg p) = 1$), so hat man in diesem Fall keine Information darüber ob p wahr oder falsch ist. Analog gilt dies für den Fall $N(p) = N(\neg p) = 0$.

Für die Disjunktion gilt $\Pi(p \vee q) = \max\{\Pi(p), \Pi(q)\}$ und für die Konjunktion $N(p \wedge q) = \min\{N(p), N(q)\}$. Dies entspricht dem intuitivem Verständnis für Possibilität und Notwendigkeit.

Gilt für eine Formel p die Ungleichung $\Pi(p) \geq N(p)$, so beuteudet dies dass die Chance dass p wahr ist mindestens so gross ist wie die Forderung dannach.

Ist für eine Formel p sowohl $N(p) > 0$ als auch $N(\neg p) > 0$, so entsteht ein Konflikt, da einerseits eine gewisse Notwendigkeit besteht dass p wahr wird und andererseits auch dass p falsch wird. Je näher $\min\{N(p), N(\neg p)\}$ dem Wert 1 kommt desto grösser wird dieser Konflikt. (Quelle: [5])

Anwendung: Erweiterung von Description Logics. Um in Description Logics auch mit Ungewissheit umgehen zu können haben Guilin Qi, Jeff Z. Pan und Qiu Ji in ihrem Paper [6] Possibilistische Logik verwendet. Für ihre neue Logik haben sie Algorithmen für einen automatischen Beweiser entwickelt und diese im KAON2-Beweiser implementiert.

2.4 Probabilistische Logik

Einer Formel wird ein Wahrscheinlichkeitsgrad zwischen 0 und 1 zugeordnet, der angibt mit welcher Wahrscheinlichkeit die Formel wahr ist.

Hat man beispielsweise das Prädikat $location(Bob, in, Room546)$ das aussagt, dass sich die Person Bob in dem Raum mit der Nummer 546 befindet, so kann hierfür die Wahrscheinlichkeit angegeben mit der es wahr ist.

$Prob(location(Bob, in, Room546)) = 0.75$ bedeutet die Wahrscheinlichkeit ist 75%.

Das Gaia Projekt. Gaia ist ein Middleware System, dass von Anand Ranganathan, Jalal Al-Muhtadi und Roy H. Campbell an der Universität Illinois entwickelt wurde. Das System dient dem Zweck Entwicklern von Ubiquitous Systemen den Umgang mit Ungewissheit zu erleichtern. Unter anderem wird dies mittels Probabilistischer Logik verwirklicht. Gaia behandelt Ungewissheit in drei grossen Gebieten: Wahrnehmung von Umwelt, Schlussfolgerungen für Kontextinformationen und Behandlung von ungewissen Kontextinformationen in Anwendungen (Quelle [7]).

Anwendungsbeispiele des Gaia Systems in Ubiquitous Computing sind die Personenlokalisierung mittels RFID-Marken und verschiedene Authentisierungsgeräte.

Anwendung RFID-Marken. Ein wichtige Anwendung ist das RFID-Marken (Radio Frequency Identification) System, dass die Identifizierung und Lokalisierung von Mitarbeitern in einem Gebäude ermöglicht. Das Gaia Projekt hat einen Lösungsansatz für die auftretende Ungewissheit mittels Probabilistischer Logik erarbeitet.

Das System basiert darauf, dass jeder Mitarbeiter eine Marke besitzt, die er zu jeder Zeit mit sich führt. Im Gebäude sind Sensoren verteilt, die erkennen welche Marken sich in ihrer Nähe befinden. Es soll festgestellt ob eine bestimmte Person sich in einem bestimmten Raum befindet. Dabei gibt es zwei problematische Ungewissheiten, die mit Hilfe der Probabilistischen Logik gelöst werden können.

Ein Faktor ist die Ungewissheit ob eine Marke sich wirklich in dem Raum befindet. Der Grund hierfür ist, dass der Wahrnehmungsbereich des Sensors über den Raum hinaus ragen kann. So können Personen erkannt werden, die aussen am Raum vorbei laufen (beispielsweise durch einen Flur). Die Wahrscheinlichkeit, dass sich die Marke wirklich im dem Raum befindet indem der Sensor ist, ist die Fläche des Raumes geteilt durch die Fläche des Dedektionskreises. Sind noch weitere Sensoren in der Nähe, so können die ebenfalls die Marke erkennen.

Ein weiterer Faktor der Ungewissheit besteht darin, dass eine Person ihre Marke nicht bei sich trägt. Beispielsweise könnte sie ihre Marke auf dem Tisch liegen lassen oder die Marke könnte gestolen worden sein. Die Wahrscheinlichkeit für diesen Fall zu bestimmen ist schwieriger, den sie variiert von Person zu Person und es ist schwierig zu verallgemeinern.

Die Wahrscheinlichkeit dass Person X im Raum Y ist, ergibt sich nun durch $Prob(location(X, in, Y)) = Prob(Marke\ von\ X\ ist\ in\ Y) * Prob(X\ hat\ seine\ Marke\ bei\ sich)$.

Authentisierungsgeräte. Viele Authentisierungsgeräte, wie Fingerdruckerkenner, Gesichts- oder Stimmerkennungssoftware, geben Grad des Vertrauens während sie Benutzer erkennen. Anhand Probabilistischer Logik wird dieser Vertrauenswert zur Authentisierung verwendet.

2.5 Evidenztheorie

Die Evidenztheorie von Dempster und Shafer ist eine Erweiterung der Wahrscheinlichkeitstheorie und wird benutzt um Informationen aus verschiedenen Quellen zu einer Aussage zu kombinieren. Dabei wird auch die Glaubwürdigkeit der Quellen berücksichtigt.

Den Anfang nimmt diese Theorie in den 60er Jahren, als Arthur Dempster begann Artikel dazu zu veröffentlichen. Glenn Shafer nahm seine Ideen auf und entwickelte sie weiter bis er schliesslich 1976 die Theorie veröffentlichte. Seitdem ist die Evidenztheorie auch unter dem Namen "The Dempster-Shafer Theory of Evidence" bekannt.

Anstatt, wie in der klassischen Wahrscheinlichkeitstheorie, Wahrscheinlichkeitswerte zwischen 0 und 1 zuzuteilen, verwendet man ein zweidimensionales Mass, das einerseits den Grad des Vertrauens (belief) in die Quelle angibt und andererseits die Plausibilität eines Ereignisses. Diese Plausibilität entspricht einen Wahrscheinlichkeitsbereich mit oberer und unterer Grenze (also ein Teilintervall des Intervalls $[0,1]$).

Um dies zu verdeutlichen, nehmen wir an dass der Wetterbericht besagt dass das morgige Wetter sonnig werden soll. Sei S das Ereignis, dass das morgige Wetter sonnig wird. Nun sei bekannt, dass der Wetterbericht mit 80%-iger Wahrscheinlichkeit recht hat. In der klassischen Wahrscheinlichkeit folgt dann aus $Prob(S) = 0.8$ dass $Prob(\neg S) = 1 - Prob(S) = 0.2$ gilt, was bedeutet dass das Wetter mit einer 20%-igen Wahrscheinlichkeit nicht sonnig sein wird. Hier unterscheidet sich die Evidenztheorie von der Wahrscheinlichkeitstheorie,

da man hier in den restlichen 20% keine Information darüber hat ob das Wetter schön sein wird, oder nicht. Der Wetterbericht hat mit Sicherheit in 80% der Fälle recht, aber in den restlichen 20% hat er nicht mit Sicherheit recht. Es kann in diesen Fällen sonnig sein oder nicht. Die Aussage ist also dass die Wahrscheinlichkeit das es morgen sonnig wird mindestens 0.8 beträgt.

Eine umfassendere Darstellung findet man in dem Buch „Schließen bei unsicherem Wissen“ ([5]).

Anwendung: UCSMdess. UCSMdess ist ein Service Model, das auf der Dempster-Shafer Evidenztheorie und auf Extended SPKI/SDSI (Symbol Public Key Infrastructure / Symbol distributed Security Infrastructure) basiert, das das Ziel hat die Sicherheit bei der Authentisierung und Identifikation in ubiquitären Systemen zu gewährleisten. Entwickelt wurde UCSMdess von Daoqing Sun, Yishu Luo und Qiying Cao, die darüber in ihrem Paper [8] berichten.

3 Kausale Modelle

Kausale Modelle modellieren Beziehungen zwischen Ursachen und Auswirkungen und sind daher hauptsächlich für diagnostische Probleme geeignet. Ich werde hier zwei Ansätze vorstellen, zuerst Bayessche Netze und dannach den relationalen Ansatz.

3.1 Bayessches Netz (Bayesian network)

Eine weitere Möglichkeit Ungewissheiten zu modellieren bieten die Bayesschen Netze. Ein Bayessches Netz ist gerichteter zyklener Graph, bei dem die Knoten Zufallsvariablen repräsentieren und die Kanten bedingte Abhängigkeiten zwischen diesen Zufallsvariablen repräsentieren.

In der Abbildung 3 ist ein Beispielgraph für ein Bayessches Netz gezeigt, das dazu geeignet ist, auf die momentan vorherrschende Aktivität im Raum zu schließen. Die Werte, die die Zufallsvariablen der Knoten annehmen können, entsprechen Kontextprädikaten. Der Knoten “Activity” repräsentiert beispielsweise ein zweistelliges Prädikat und $activity(x, y)$ gibt an ob im Raum x gerade die Aktivität y im Gange ist (Beispiel für Raum Nummer 2401: $activity(room2401, meeting)$, $activity(room2401, presentation)$, $activity(room2401, idle)$).

Um die Wahrscheinlichkeit einer bestimmten Situation zu berechnen, beispielsweise dass keine bestimmte Aktivität herrscht und keine Leute in dem Raum sind usw., kann als Produkt bedingter Wahrscheinlichkeiten berechnet werden ($Prob(activity(idle)s, NumOfPeopleInRoom(0), LightStatus(Off) \dots) = Prob(activity(idle) \mid NumOfPeopleInRoom(0)) * Prob(activity(idle) \mid LightStatus(Off)) * \dots$).

Anwendung: Erkennung von ungewöhnlichen Ereignissen in ubiquitären Sensorumgebungen. Die NTT Network Innovation Laboratories in Tokyo

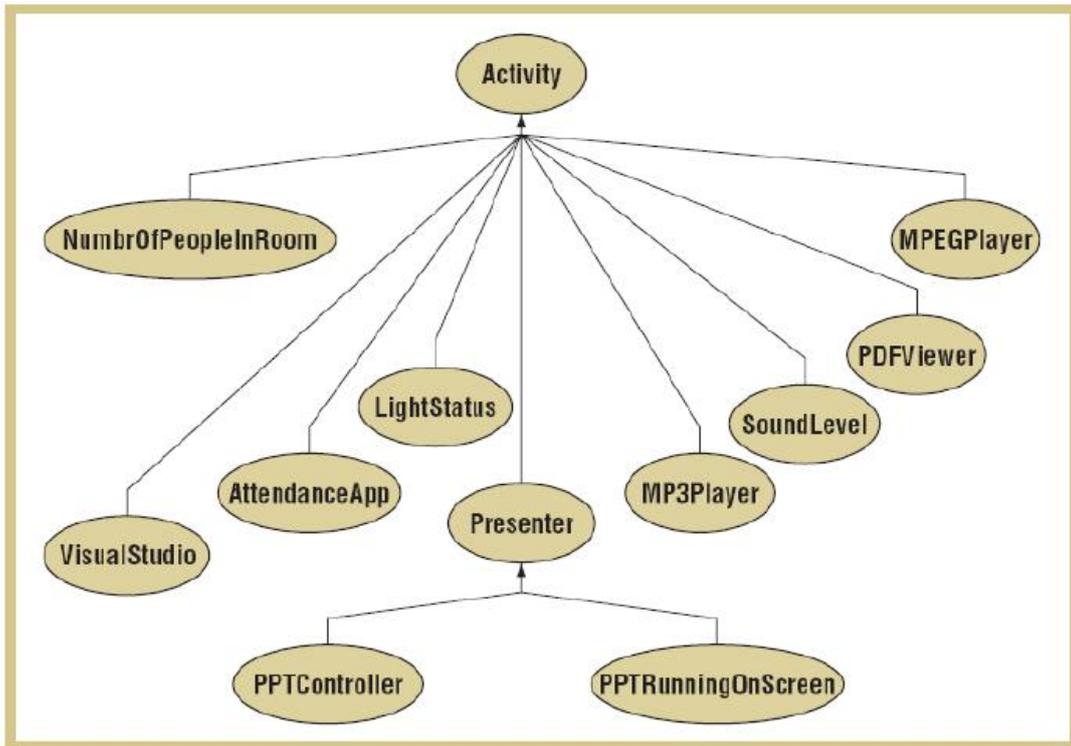


Abb. 3. Beispiel eines bayesschen Netzes

haben einen Ansatz ausgearbeitet um nicht vorgesehene Ereignisse in Sensorumgebungen zu mit Hilfe von Bayesschen Netzen zu erkennen. Sun Yong Kim, Miyuki Imada und Masakatsu Ohta haben in dem Paper [9] diesen Ansatz vorgestellt.

Das System dient dazu ungewöhnliche Ereignisse in Wohnungen zu erkennen, da in solchen Umgebungen das Auftreten von hervor nicht berücksichtigten Ereignissen besonders gross ist. Dies liegt daran, dass das Umgebungsverhalten in menschlichen Lebensräumen sehr komplex ist. Beispielsweise können durch das Verhalten der Bewohner oder Besucher unvorhergesehen Ereignisse eintreten.

Abbildung 4 zeigt ein bayessches Netz, das ein Sensornetzwerk zu verschiedenen Zeitpunkten modelliert. Angenommen es gibt p Sensoren S_1, \dots, S_p in einem Sensornetzwerk. Ein Sensor S_i wird hierbei als Zufallsvariable betrachtet. Dabei sei der Sensor S_i abhängig von allen Sensoren mit einem kleiner Index, also S_1, \dots, S_{i-1} . In den bayesschen Netz werden diese Abhängigkeiten als Kanten zwischen den Knoten, die den Zufallsvariablen entsprechen, repräsentiert. Die Zufallsvariable x_{ij} beschreibt nun das Ergebnis des Sensors S_j zum Zeitpunkt t_i . Dann ist $X_i = (x_{i1}, \dots, x_{ip})^T$ ein Zufallsvektor, der die Informationen der Sensoren zum Zeitpunkt t_i wiedergibt. Ein Vektor X_i ist nun nur vom direkten Vorgänger X_{i-1} abhängig. Dies wird in der Abbildung übersichtlich dargestellt.

Dieses bayessche Netz kann nun dazu dienen ungewöhnliche Ereignisse festzustellen. Für eine genauere Darstellung siehe [9].

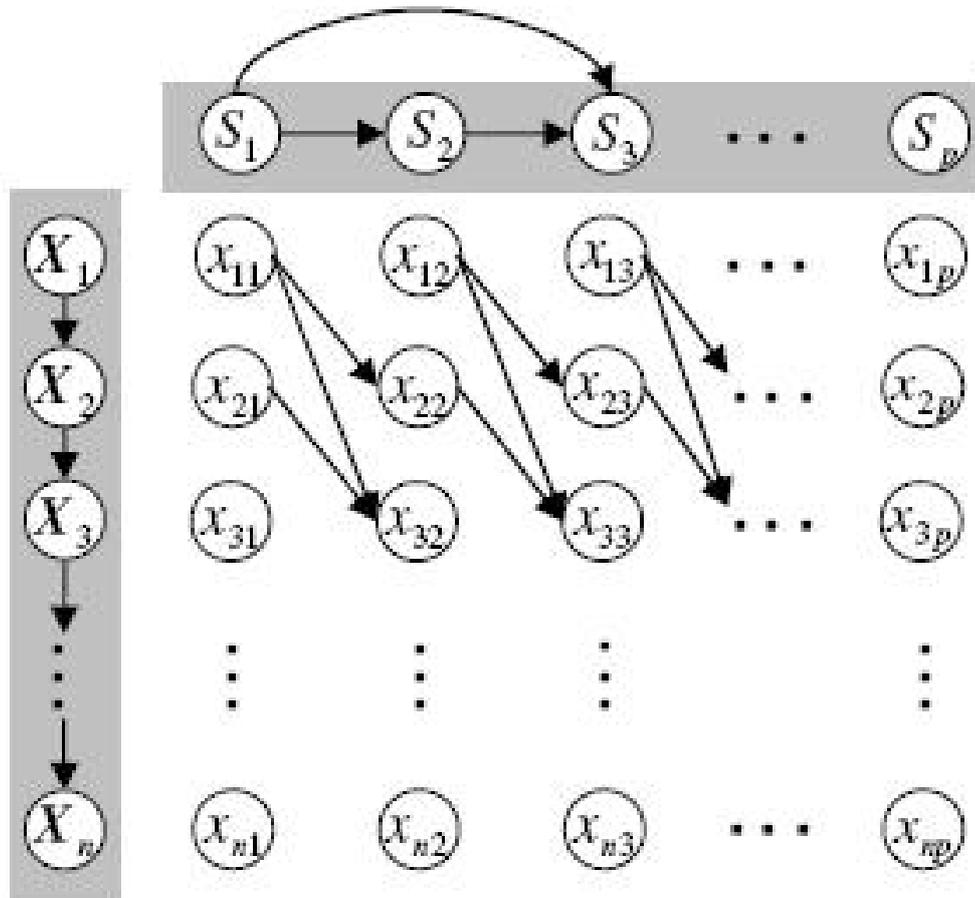


Abb. 4. Bayessches Netz zur Erkennung von ungewöhnlichen Ereignissen

3.2 Der relationale Ansatz

Ein weiteres Kausalmodell um Beziehungen zwischen Ursachen und Wirkungen unter Berücksichtigung von Unsicherheit zu modellieren ist der relationale Ansatz.

Ein Modell nach dem relationalen Ansatz besteht aus einer Menge von beobachteten Ereignissen E , einer Menge von möglichen Schlüssen G und einer Relation R zwischen E und G . Steht ein Ereignis $e \in E$ mit einem Schluss $g \in G$ in Relation, so bedeutet dies, dass der aus der Ursache g das Ereignis e ist. Eine umfassendere Darstellung findet man in dem Buch „Uncertainty based information“ ([3]).

4 Ontologien mit Ungewissheit

Ontologien dienen dazu, Wissen zu strukturieren indem Entitäten und Beziehungen zwischen diesen Entitäten modelliert werden. Ein wichtiger Anwendungs-

bereich ist die Web Semantik, die das Ziel verfolgt Webseiten eine Bedeutung zuzuordnen. Informationen zur Web Semantik findet man in [10] und [11].

Mit Ontologien kann man eine Welt klassifizieren und beschreiben. Ontologien bestehen aus Klassen, binären Relationen zwischen den Klassen, Formale Axiomen um die Konsistenz zu sichern und Objekten.

Beispiel.

- Klasse „Student“ und „Professor“:
 $\forall x (Student(x) \rightarrow \exists y (ingeschrieben(x, y) \wedge Kurs(y)))$
 $\forall x (Professor(x) \rightarrow \exists y (unterrichtet(x, y) \wedge Kurs(y))).$
- Relation „ingeschrieben“:
 $\forall x, y (ingeschrieben(x, y) \rightarrow Student(x) \wedge Kurs(y)).$
- $\forall x Professor(x) \wedge Student(x).$
- Student(Lea).

Yi Yang and Jacques Calmet von der Universität Karlsruhe haben dargestellt in [12], wie man ein Bayessches Netz in Ontologien integrieren kann. Dadurch werden die Vorteile beider Ansätze kombiniert.

In vielen Anwendungen von Ubiquitous Computing ist es schwer eine eindeutige Klassifikation der Welt zu finden. Dies liegt an den Differenzen zwischen der Sprache der Benutzer und der des Systems. Die natürliche Sprache ist vage und die Bedeutung ist nicht immer eindeutig.

5 Fazit

In der Gesamtheit der hier vorgestellten Methoden soll versucht werden, ein repräsentatives Panorama des momentanen Stands der Kunst über die Verarbeitung und Erzeugung von ungewissen und unscharfen Informationen in ubiquitären Systemen zu geben.

In ubiquitären Systemen ist die Behandlung von ungewissen und unsicheren Informationen eine wichtige Aufgabe. Die hier vorgestellten Methoden ermöglichen die Zuverlässigkeit und Genauigkeit dieser Systeme zu verbessern. Damit Benutzer diesen Systemen Vertrauen schenken, gerade in sicherheitsrelevanten Gebieten, ist es wichtig einen hohen Grad der Zuverlässigkeit zu gewährleisten.

Die Unscharfe Logik bietet die besten Möglichkeiten Ungenauigkeit oder Vagheit zu behandeln. Gerade weil in ubiquitären Systemen das Wissen über die Umgebung eine große Rolle spielt und diese Kontextinformationen mittels Sensoren gesammelt werden, bietet sich die Unscharfe Logik an um die Sensordaten besser in die Informationsverarbeitung einfließen zu lassen. Allerdings stößt man bei dieser Methode an die Grenzen, wenn es um Wahrscheinlichkeiten geht.

Will man Wahrscheinlichkeiten oder Unsicherheiten in dem Informationsverarbeitungsprozess berücksichtigen, so gibt es verschiedene Möglichkeiten. Die naheliegendste ist wohl die Probabilistische Logik, bei der man einfach mit Wahrscheinlichkeiten rechnet. Ein wenig komplexere Ansätze sind die Evidenztheorie oder die Possibilistische Logik.

In ubiquitären Systemen stellt sich häufig das Problem, dass das System auf gewisse Ereignisse reagieren soll. Dies kann als Beziehung zwischen Ursachen und Wirkungen modelliert werden, hierfür bieten sich beispielsweise bayessche Netze an.

Wie ich in dieser Seminararbeit gezeigt habe, gibt es für die Verarbeitung von ungewissen und unscharfen Informationen in ubiquitären Systemen viele verschiedene Ansätze. Es liegt an der konkreten Problemstellung welchen dieser Ansätze man verfolgt, oder ob man möglicherweise eine Kombination mehrerer Methoden wählt.

Literatur

1. Patrick Blackburn, J.v.B., Wolter, F.: Handbook of modal logic. <http://www.csc.liv.ac.uk/~frank/MLHandbook/>
2. Schmitt, P.: Formale systeme. <http://i12www.ira.uka.de/%7Ebeckert/Lehre/Formale-Systeme/skriptum.pdf>
3. George J. Klir, M.J.W.: Uncertainty-Based information. Elements of Generalized Information Theory. Physica-Verlag
4. Mihai Marin-Perianu, Clemens Lombriser, Oliver Amft, Paul Havinga, Gerhard Tröster: Distributed activity recognition with fuzzy-enabled wireless sensor networks. <http://www.springerlink.com/content/ut1u651151372710/fulltext.pdf>
5. Sombé, Léa: Schließen bei unsicherem Wissen in der künstlichen Intelligenz : Vergleich von Formalismen anhand eines Beispiels . Braunschweig [u.a.] : Vieweg, 1992. - VII, 181 S. : graph. Darst.; (deutsch) (Künstliche Intelligenz)
6. Guilin Qi, Jeff Z. Pan, Qiu Ji: Extending Description Logics with Uncertainty Reasoning in Possibilistic Logic. <http://www.springerlink.com/content/j5102m4034j04235/fulltext.pdf>
7. Schmitt, P.: Skriptum nicht klassischke logiken. <http://i12www.ira.uka.de/~pschmitt/NKL/NKLFolien.pdf/31L.pdf>
8. Daoqing Sun, Yishu Luo, Qiyang Cao: UCSMdes: Ubiquitous Computing Service Model based on D-S Evidence Theory and Extended SPKI/SDSI. <http://www.naun.org/journals/computers/ijcomputers-25.pdf>
9. Sun Yong Kim, Miyuki Imada, Masakatsu Ohta.: Detecting Anomalous Events in Ubiquitous Sensor Environments using Bayesian Networks and Nonparametric Regression. <http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=04220899>
10. Mauro Mazzieri, A.F.D.: A fuzzy semantics for semantic web languages. <http://ftp.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-173/paper2.pdf>
11. Hitzler, P., Krötzsch, M., Rudolph, S., Sure, Y.: Semantic Web Grundlagen. <http://www.springerlink.com/content/978-3-540-33993-9>
12. Yi Yang, J.C.: Ontobayes: An ontology-driven uncertainty model. http://avalon.ira.uka.de/iaks-calmet/papers/IAWTIC05_yang.pdf
13. Anand Ranganathan, Jalal Al-Muhtadi, R.H.C.: Reasoning about uncertain contexts in pervasive computing environments. <http://choices.cs.uiuc.edu/~ranganat/Pubs/b2ran%5B1%5D.lo.pdf>
14. Schmidt, A.: Kontext-middleware zur verwaltung dynamischer und unvollkommener kontextinformationen. <http://i12www.ira.uka.de/~pschmitt/NKL/NKLFolien.pdf/31.pdf>.

15. Schmidt, A.: Kontext-middleware zur verwaltung dynamischer und unvollkommener kontextinformationen. <http://wwwbayer.in.tum.de/aktuelles/GI-Workshop/themen/schmidt.pdf>

Seminarausarbeitung Simulation von Ubiquitären Systemen

Adrian Genaid

Telecooperation Office

1 Einleitung

Ubiquitäre Systeme beinhalten ein weites Feld an Eigenschaften und Erfordernissen. Ihre Entwicklung und Evaluierung ist aufgrund hoher Heterogenität in Hardware und Software, einer großen Anzahl solcher Geräte sowie der Kontextsensitivität sowohl sehr zeitaufwändig als auch teuer. Manche Plattformen sind ausserdem noch nicht in gewünschter Form, Größe, Rechenleistung oder Sensorausstattung erhältlich, so daß passende Geräte zunächst einmal selbst entworfen beziehungsweise bestehende erweitert werden müssten (Vgl. [3]).

Desweiteren sind solche Systeme nur schwer zu testen, denn aufgrund der vielfältigen Kombinationen eines kontextsensitiven Systems, welches durch die Interaktion von Nutzern in Form von Mobilität oder anderem seinen Sinn erhält, ist es nicht möglich, viele Kontextkombinationen in der Realität auf Funktionalität zu prüfen.

Ein Simulationswerkzeug für Ubiquitäre Systeme gibt dem Entwickler oder Forscher die Möglichkeit, simulierte Szenarien mit simulierten Plattformen durchzuspielen, um so mit wesentlich geringerem Aufwand vorab entscheiden zu können, ob das System seinen Anforderungen gerecht werden kann.

1.1 Simulationsziele

Allgemein wird Simulation überall dort eingesetzt, wo unter Realbedingungen durchgeführte Evaluationen ineffektiv oder nicht möglich sind und eine formal beschreibbare Verifikation nur mit nicht vertretbarem Aufwand erreichbar ist. Wie oben beschrieben, ist dies häufig der Fall bei verteilten Anwendungen, bei denen Geräte, Software und Kommunikationsschnittstellen mit unterschiedlichsten Ausprägungen ein Gesamtsystem bilden. Außerdem spielt Simulation auch dort eine Rolle, wo zu bestehenden Systemteilen weitere hinzugefügt werden sollen.

1.2 Simulation Ubiquitärer Systeme

Das breite Feld des Ubiquitous Computing bringt viele Herausforderungen an ein Simulationswerkzeug mit sich. Es sind viele Simulatoren entstanden, die von einem Feld an die jeweiligen Bedürfnisse der Forschungsgruppe angepasst wurden, so daß selten alle Felder einerseits ausreichend detailgetreu und andererseits generell genug simuliert werden können [3].

In diesem Artikel werden zunächst die Voraussetzungen zur Beschreibung und Simulation Ubiquitärer Systeme beschrieben, dann werden verschiedene ausgewählte Simulationswerkzeuge beschrieben und deren Eignung für die genannten Voraussetzungen diskutiert.

2 Voraussetzungen für Ubiquitous Computing Simulationen

Ubiquitous Computing, als ein weites und relativ junges Forschungsfeld, wirft Entwicklungsfragen in vielen Domänen auf, ähnlich wie Konzepte aus der Mainframeära beim Übergang zum Personal Computing vollkommen überarbeitet werden mussten. (Vgl. [1] und [2])

Ubiquitäre Systeme erfordern somit neue Entwicklungsansätze, unter anderem von Hardwarekomponenten (vor allem hinsichtlich Energieverbrauch und Sensorik), drahtlosen Netzwerktechnologien wie zum Beispiel dem WPAN, Geräteformfaktoren, Netzwerkprotokollen (zum Beispiel Unterstützung von „Mobile IP“), Kontextsensitive/Kollaborative Applikationen verschiedenster Art und anderes.

Um solche Ansätze – die am Allgemeinen nicht mit angemessenem Aufwand im mathematischen Sinne verifizierbar sind – evaluieren zu können, sind Simulationswerkzeuge unverzichtbar, denn auch wenn Simulation nicht sicherstellt, daß das fertige „Produkt“ wirklich funktioniert, können falsche Entwicklungsschritte erkannt und vermieden werden.

Generell arbeitet Simulation durch Abbildung von der Realität auf ein Modell und Erprobung des Problems auf diesem Modell. Ein Simulator bietet vordefinierte Modelle, die innerhalb von vordefinierten Grenzen verfeinert werden können, und dann zum Zwecke der Evaluation aufeinander angewendet werden.

2.1 Generelle Voraussetzungen für Simulationssoftware

Ein Simulator als ein Werkzeug ist für große Teile von an einem Feld Forschenden nur dann interessant, wenn das Verhältnis zwischen Aufwand und erhaltenem

Ergebnis für möglichst viele stimmt. Nach [3] ist dies der Fall, wenn die folgenden Eigenschaften zutreffen:

Flexibilität Simulatoren sind relativ speziell für verschiedene Forschungsgebiete. Dennoch gibt es in einem etablierten Forschungsfeld meist nur sehr wenige weitverbreitete Simulatoren. Dies lässt sich daraus erklären, daß solche Simulatoren generell genug sind, um für möglichst viele Gruppen benutzbar zu sein.

Benutzbarkeit (Usability) Zu generelle Simulatoren tendieren jedoch zu einer Unbenutzbarkeit, da aussagekräftige Simulationen nur durch langwierige Anpassung der Modelle/Entwicklung von großen Teilen der Software erreicht werden können. Also ist es notwendig, den Weg zwischen hoher Flexibilität und Benutzbarkeit zu finden. Eine recht gute Vorhergehensweise ist es, dem Nutzer des Simulators die Freiheit über den Grad der Verfeinerung der Modelle zu geben, so daß dieser schneller Ergebnisse für die für ihn relevanten Teile bekommt.

2.2 Extraktion von Anforderungen für Ubiquitous Computing Simulationen

Um klären zu können, ob ein Simulationswerkzeug für die Simulation von ubiquitären Systemen geeignet ist, muß zunächst dargestellt werden, was in diesem Feld simuliert werden muss, um aussagekräftige Ergebnisse zu erhalten.

Die Beschreibung von „Ubiquitous Computing“ nach [2] selbst genügt nicht, um die Erfordernisse für ein Modell aufzuzeigen, beziehungsweise ist zu allgemein, um daraus ein Modell für die Simulation entwickeln zu können. Jedoch lassen sich aus den inzwischen zahlreichen, bestehenden Applikationen und beschriebenen, noch nicht real existierenden Applikationen typische Anforderungen gewinnen, mit deren Hilfe Simulationsumgebungen auf ihre Eignung für das Feld des Ubiquitous Computing überprüft werden können.

Somit werden nun zunächst exemplarisch typische Anwendungen kurz beschrieben und in die Einzelanforderungen bezüglich Kontextbewusstsein zerlegt. Die Beschreibungen stammen aus [7]. Danach werden diese so generell wie nötig zusammengefasst als allgemeine Anforderungen von Ubiquitous Computing Applikationen.

Cyberguide ist in mobiles, kontextbewusstes Tourführungssystem im GVV-Zentrum der Georgia Tech. Hierin werden Ort, die Bewegungsrichtung und die bereits besuchten Orte genutzt, um andere für den Benutzer möglicherweise interessante Orte vorzuschlagen.

Beim *Active Badge Location System* tragen Benutzer Erkennungszeichen, die Signale versenden. Diese Signale werden von einem Sensornetz aufgenommen

und an einen Lokalisierungsservice weitergereicht. Hierdurch ist Lokation einer Person möglich.

Der *Conference Assistant* hilft Konferenzteilnehmern durch automatische Anzeige des Konferenzplans und zeigt mehrere Interessenpfade auf, denen der Nutzer folgen kann. Das System führt den Nutzer auch zu Veranstaltungsorten, an denen der für ihn interessante Pfad abgehalten wird.

Als eines der wichtigsten Kriterien kann sich für einen Leser aus dieser Aufzählung die Lokation herausstellen. Jedoch genügt nur dieses Kriterium bei weitem nicht, denn ein Ort wird oft erst zusammen mit anderen Merkmalen zu einer für den Benutzer wichtigen Information. Zum Beispiel könnte in einer Touristensituation der Aufenthaltsort dafür verwendet werden, die Sehenswürdigkeiten in der Nähe anzuzeigen, oder bei Magenknurren naheliegende Restaurants, die regionale Spezialitäten anbieten. Nach [7] wird Kontext in drei Arten aufgeteilt:

- Computing Contexts
- User Context
- Physical Context

In Tabelle 1 werden verschiedene Kontexte in diese Kategorien einsortiert.

In [3] werden Ubiquitous-Computing-Szenarien in vier Abstraktionen aufgeteilt, welche ausreichen, um diese zu beschreiben. Diese Abstraktionen sind:

- Umgebung
- Sensoren
- Aktoren
- Applikationen

Hierdurch ist der Benutzerkontext für Simulationen keine eigene Kontextart mehr, da auch dies eine physische Einwirkung auf die Umgebung darstellt, also stellt der Benutzer eines Gerätes einen Akteur dar, dessen Phänomen von einem Sensor des Gerätes gemessen wird.

Tabelle 1. Classification of some typical Contexts into Categories, taken from [7]

Computing Contexts	User Contexts	Physical Contexts
Available Devices	Preference	Location
CPU	Purpose	Time
Memory	User Calendar	Destination
Screen size	Personal information	Traffic Condition
Energy	Facilities	Physical limitations
Bandwidth	Disability	Weather

3 Evaluation verschiedener Simulatoren

Zur Simulation gewisser Teilaspekte von ubiquitären Anwendungen können, wie in Abschnitt 1 beschrieben, viele grundverschiedene Simulatoren verwendet werden. Viele verwendete Simulationstools stammen aus dem Bereich der Computernetzwerk-Simulatoren und wurden – meist nach Anpassung auf Sensornetze – auf die Bedürfnisse einzelner Anwendungen angepasst. Dies liegt nahe, denn auch wenn ubiquitäre Systeme in keinster Weise auf die Netzwerkkommunikation beschränkt sind, ist dies eine Kernkomponente, da Ubiquitous Computing nach [2] viele verschiedenartige Rechner verwendet, die vernetzt die Umgebung erweitern.

Manche Simulationstools jedoch wurden mit vollkommen entgegengesetzten Zielen entworfen, diese konzentrieren sich eher auf den Aspekt der Benutzbarkeit von Geräten und ubiquitären Umgebungen durch Menschen, siehe zum Beispiel [8]. Auch wenn in manchen der im nachfolgenden vorgestellten Simulatoren ein ähnliches Konzept in Ansätzen vorhanden ist, so sollen solche Simulatoren in diesem Artikel nicht behandelt werden aus im folgenden genannten Gründen nicht behandelt werden.

3.1 Einschränkungen durch grundsätzliche Eigenschaften

Zunächst einmal scheiden zu dem oben genannten Zweck die ziemlich früh, aus der Ubiquitous Computing-Thematik heraus entstandenen Simulatoren aus, die sich rein auf den Akzeptanz- und Benutzbarkeitsaspekt von Anwendungen beschränken. Diese Simulatoren arbeiten häufig in 3D/Virtual Reality-Umgebungen, benötigen also ebenso Nutzeraktivität wie in realen Tests. Dadurch sind große/viele Testläufe in ubiquitären Umgebungen kaum realisierbar. Simulatorbeispiele sind UbiWise [8], QuakeSim [9] usw.

Desweiteren sollen reine Sensornetzwerksimulatoren ausgeschlossen werden. Hier von gibt es einerseits solche, die für einen Typ von Sensorknoten ausgelegt sind, wie zum Beispiel Tossim. Diese haben meist zusätzlich zu der Beschränkung der Hardwareplattform noch die Fixierung auf ein Image, welches auf allen Knoten installiert ist. Ein zweiter Typ von Sensornetzwerksimulatoren baut meist auf einem der generellen Computernetzwerksimulatoren wie ns-2 oder OMNet++ auf und hat oft Ähnlichkeiten mit solchen, die für Ubiquitäre Systeme verwendet werden. Jedoch beschränken sich Simulatoren dieses Typs meist rein auf die Belange von Sensorknoten, wodurch sich das Verständnis von Umgebung auf die Hindernisse für Funksignale beschränkt. Zu nennende Beispiele hier sind Sens [11] und Mixim [10].

3.2 Betrachtung der Simulatoren in der eingeschränkten Auswahl

VisualSense ist im Rahmen des Ptolemy Project entstanden, worin Modellierung, Simulation und Design von nebenläufigen eingebetteten Echtzeitsystemen erforscht werden (Vgl. [12]). Es ist ein Modellierungs- und Simulationsframework für Sensornetzwerke und drahtlose Netzwerke, welches eine Spezialisierung von Ptolemy II darstellt (siehe [6]). Es ist in der Ptolemy II Distribution bereits enthalten.

Ptolemy II ist in der Programmiersprache Java geschrieben, es läuft also auf einer Vielzahl von Systemen. Desweiteren lassen sich auf diese Art die Modelle von Komponenten auch mit Java realisieren. Eine Stärke an Ptolemy ist jedoch, daß Komponenten aus bereits erstellten Komponenten über eine graphische Benutzerschnittstelle intuitiv zusammengestellt und verbunden werden können.

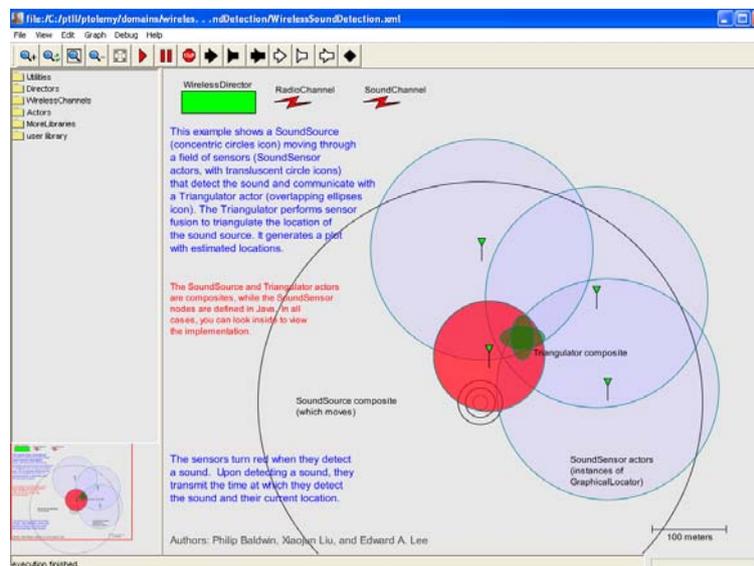


Abbildung 1. Beispielsimulation für die Triangulation einer Tonquelle in Ptolemy/Visualsense

Es fällt bei der Benutzung des Systemes auf, daß die Komponenten aus mehreren Subsystemen bestehen. Diese Subsysteme wiederum kommunizieren untereinander über Ports, wodurch eine starke Ähnlichkeit zu Hardwarebeschreibungssprachen wie VHDL oder auch Programmierdialekten wie das von TinyOS verwendete NesC besteht. Die einzelnen Komponenten sind teilweise recht abstrakt, dennoch erinnert das Blockschaltbild (siehe Abbildung 2) eines Modells stark an die Abstammung aus Elektrotechnik/Echtzeitsystemen.

Grundätzlich implementieren Komponenten das *Actor*-Interface, diese sind entweder *atomic* und damit in Java definiert, oder *composite*, d.h. sie bestehen aus

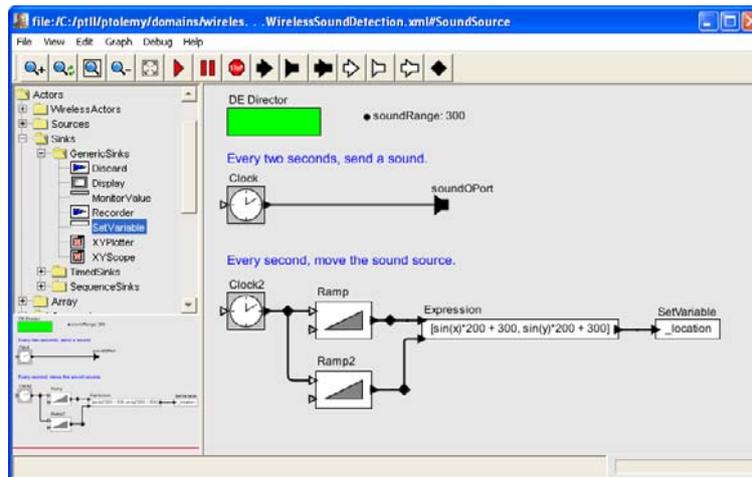


Abbildung 2. Blockschaltbild einer sich kreisförmig bewegenden Tonquelle in Ptolemy/Visualsense

anderen *Actor*-Instanzen und Relationen unter diesen. Ptolemy bietet eine recht umfangreiche Library an Aktoren an. Im folgenden werden manche beschrieben werden, die für Ubiquitäre Systeme eine Rolle spielen.

Durch einen Discrete-Event-Mechanismus werden Events zwischen Komponenten mit Hilfe eines Event-Timestamps von einem Scheduler nacheinander abgearbeitet. Komponenten besitzen in Ptolemy II wie in anderen Discrete-Event-Simulationstools Ports, über die sie untereinander verbunden werden. In VisualSense jedoch ist aufgrund der Ausprägung der drahtlosen Kommunikation dieser Verbindungsmechanismus durch einen Kanal erweitert worden, der zwischen mehreren Ports liegt. Damit wird der Mobilität und Änderung der Umgebung (zum Beispiel temporäre Hindernisse) von Sensorknoten und anderen drahtlosen Kommunikationsteilnehmern Sorge getragen.

Durch das objektorientierte Paradigma von Ptolemy können verschiedenste Verfeinerungen der Grundmodelle in VisualSense vorgenommen werden. Eine Nachricht von einem Knoten wird im Standardverfahren an alle anderen im Kanal übertragen, während sich dieses Modell leicht so anpassen lässt, daß über eine Entfernungsmessung zwischen dem sendenden Knoten und jedem möglichen Empfänger die Nachricht nur an diejenigen geht, die in einem bestimmtem Bereich sind. Außerdem kann auch über Geländemodelle das Kommunikationsmodell verfeinert werden. Durch dieses Verfahren ist die in Abschnitt 2.1 beschriebene „Benutzbarkeit“ gegeben, da dem Benutzer freigestellt wird, wie weit er sein Modell verfeinern möchte.

Über die Verschaltung von in der Ptolemy-Library vorhandenen Aktoren wie *Clock* und *Ramp* lässt sich ein Bewegungsmodell erstellen. Außerdem gibt es bereits sogenannte *HigherOrderActors*, unter denen sich ein mobiler Actor befindet. So könnten beispielsweise von Nutzern getragene oder sich in Fahrzeugen

befindliche Kommunikationsgeräte modelliert werden, die mit anderen, stationären oder ebenfalls mobilen Geräten kommunizieren sollen. Außerdem gibt es noch den Aktor *Locator*, welcher die momentane Position zurückliefert.

Auf den ersten Blick beschränkt sich die Umgebung auf Hindernisse, dennoch kann man mit VisualSense sehr viel mehr erreichen. Dem Nutzer wird aufgrund der hohen Generalität des Simulationssystems und der einfachen Erweiterung der Bibliotheken ein unerschöpflicher Spielraum gegeben. Ein Beispiel aus [6] zeigt eine Tonquelle, welche durch Triangulation lokalisiert wird. Hierzu mussten die Grundmodelle jedoch stark angepasst werden. Dies ist ein großer Nachteil an VisualSense, denn auch wenn Usability gegeben ist, solange man sich weitgehend an das eigentliche Einsatzziel von VisualSense hält und somit auf viele der bereits entwickelten Komponenten zurückgreifen kann, führen komplexere, neue Modelltypen, wie sie bei der Entwicklung von Ubiquitären System zwangsläufig entworfen werden müssten, zu hohem Entwicklungsaufwand.

pythonDEVS ist eine Pythonimplementierung von DEVS, der *Discrete Event Specification*. Diese bietet eine Möglichkeit, wie der Name schon sagt, diskrete Ereignisse zu modellieren. DEVS ist keinem bestimmten Simulationsfeld zuzuordnen; alles, was mit Hilfe diskreter Ereignisse modelliert werden kann, lässt sich mit Hilfe von DEVS modellieren (siehe hierzu [4]).

Hier sind Komponenten, ähnlich der Vorhergehensweise von Ptolemy, entweder Instanzen der Basisklassen, *atomic* genannt, oder strukturelle Zusammenschlüsse, sowohl von atomaren als auch gekoppelten Komponenten, die als *coupled* bezeichnet werden (siehe Abbildung 4). Auch die auf Ports aufbauende Verbindungslogik haben DEVS und Ptolemy gemein.

Jedoch ist der DEVS-Formalismus gerade im Aufbau von *Atomic-DEVS*-Instanzen klar definiert: ein Atomic Model AM ist definiert wie in Abbildung 3 angegeben. X repräsentiert die Eingabedaten von AM , Y ist die Ausgabe. Ausserdem besitzt AM eine während der Simulation abfragbare Statusvariable S . Diese wiederum kann von den Funktionen δ_{ext} , δ_{int} , λ und t_a manipuliert werden. Die externe Überföhrungsfunktion δ_{ext} wird bei Auftreten eines Events auf den Input-Ports des Modelles ausgeföhrt, die Ausgabefunktion λ wird zur Ausgabe des Ergebnisses verwendet, die interne Überföhrungsfunktion δ_{int} wird nach Übergabe der Ausgabedaten Y ausgeföhrt. Schließlich wird t_a zur zeitabhängigen Veränderung des Status verwendet.

Hieraus kann, wie in Artikel [4] beschrieben, auf relativ einfachem Weg ein Sensorknoten beschrieben werden. Dieser besteht aus den Einzelaktoren Kommunikationsmodul, Prozessor, Speicher, Batterie, Sensoren. Verbindung zur Außenwelt wird hergestellt über ein eigenes atomares Modell, welches das modellierte Sensorboard mit Daten versorgt. In der konkreten Aufgabe der Autoren des Artikels, eine Früherkennung von Waldbränden zu entwickeln, wurde das Hauptaugenmerk auf Temperatur gelegt. Aufgrund des allgemeinen Charakters von

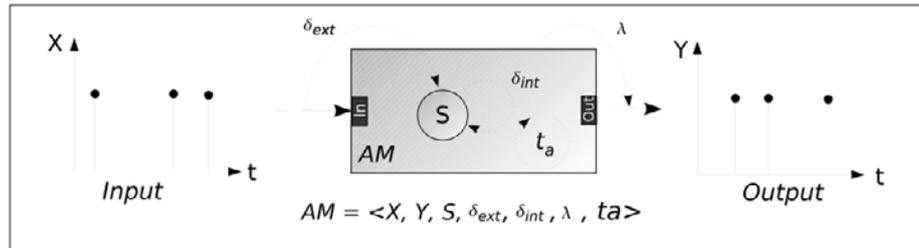


Abbildung 3. Aufbau des Atomic Models in DEVS

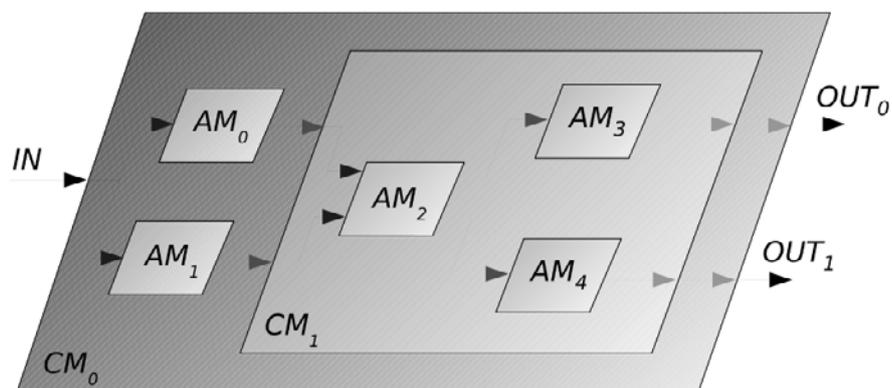


Abbildung 4. Aufbau des Coupled Models in DEVS

DEVS macht es jedoch keinen Unterschied, welche Art von Daten hier an die Sensoren geliefert wird.

Die in Abschnitt 2.1 beschriebene, als Flexibilität bezeichnete Eigenschaft besitzt DEVS in ziemlich hohem Maße. Interessant ist auch, daß DEVS automatisch eine Simulation für jedes Modell bietet. Anders als beispielsweise Ptolemy oder Netzwerksimulatoren wie ns-2 jedoch sind keine Modelle vorgegeben, die nun verfeinert werden können. Ein Modell für die Übertragungsleistung des Kommunikationsmoduls beispielsweise müsste experimentell/mathematisch entwickelt werden. Allerdings ist es auch einfach, eine Bibliothek an Modellen mit anderen Forschungsgruppen gemeinsam aufzubauen, auch wenn dies bisher scheinbar in nur geringem Maße geschehen ist.

PythonDEVS bietet keine graphische Benutzeroberfläche wie VisualSense/Ptolemy, auch die Simulation erzeugt eine Textausgabe, die schwierig zu interpretieren ist. Jedoch existiert ein graphischer Aufsatz zur Modellerstellung namens Atom, welcher die Benutzerfreundlichkeit erhöht.

Siafu [5] ist im Verlauf des IST Projektes MobiLife von den NEC-Laboratories entwickelt worden. Dieses Projekt beschäftigte sich mit der Erforschung und Entwicklung mobiler Applikationen, bei denen die Nutzer im Mittelpunkt standen und wodurch praxisnahe Applikationen entstehen sollten. Siafu wurde in der Programmiersprache Java entwickelt, anders als die beiden zuvor vorgestellten Simulatoren folgt dieser einem Agentenprinzip, das ein wenig an das Konzept von UbiWise erinnert.

Siafu baut vor allem auf der Umgebung auf, der Kontext ist das beherrschende Paradigma. Der erste Schritt in der Nutzung des Simulators ist die Definition einer Umgebung durch Karten im PNG-Format. Hierzu wird zunächst ein Hintergrundbild erstellt, auf diesem aufbauend definiert man dann die begehbaren/befahrbaren Flächen durch ein Hindernisbild (*wall image*), siehe hierzu Abbildung 5. Nun kann man Orte definieren, dies kann auf zwei Arten geschehen: Entweder werden im *WorldModel* Koordinaten für die Orte angegeben, oder es wird eine weiße Karte erstellt, welche für jeden Ort einen schwarzen Pixel erhält. Die zweite Vorhergehensweise wird von den Entwicklern empfohlen, da die Koordinatenberechnung nervraubend sein kann.

Kontextvariablen können entweder einem Ort zugeordnet werden (diese heißen dann *Overlays*), oder aber einem Agenten. Overlays können wiederum im *ContextModel* definiert werden, die Autoren raten jedoch zur Nutzung von Bildern zur Initialisierung, wie dies in Abbildung 6 dargestellt wird. Es gibt binäre Overlays, wodurch die Variable für einen Pixel basierend auf einem Schwellwert *wahr* oder *falsch* annimmt, oder es können mehrere Schwellwerte definiert werden zur Belegung einer Variable mit einem bestimmten Wert, oder aber der Pixelwert ist gleichzeitig der Wert der Variable.

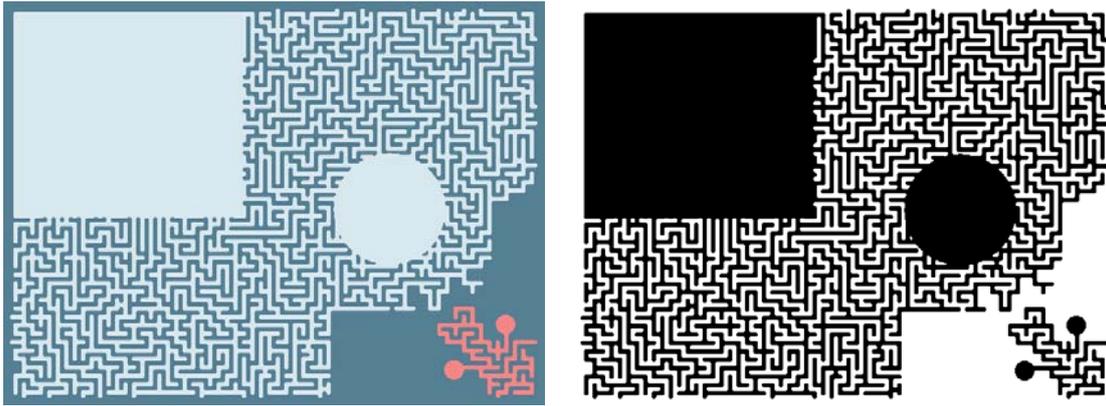


Abbildung 5. Hintergrund und begehbare Flächen in Siafu



Abbildung 6. Verschiedene Overlays in Siafu

Nun kann die Programmierung des Szenarios beginnen, hierzu müssen drei Verhaltensmodelle implementiert werden:

- *AgentModel*: Verhaltensbeschreibung, Mobilitätsmodell
- *WorldModel*: Aktionen an den Orten
- *ContextModel*: Entwicklungsbeschreibung der Overlays

Diese Modelle werden durch Implementierung der Klassen *BaseAgentModel*, *BaseWorldModel* und *BaseContextModel* angepasst. Hier können die zwei Methoden *createX* und *doIteration* überschrieben werden. Die create-Methode wird beim Erstellen der Welt aufgerufen, daher können hier die Agenten, Orte und Overlays erzeugt werden. *doIteration* wird bei jeder Iteration durchlaufen und gibt somit die Möglichkeit, Verhaltens-/Umgebungsänderungen durchzuführen.

Der Fokus dieses Simulators liegt eindeutig auf Menschen, die in einer Umgebung agieren, ohne daß dies zwangsläufig eine Ubiquitous Computing Umgebung ist. Ein Beispiel für eine Umgebung ist das Büroszenario in Abbildung 7. Es ist auch die Möglichkeit enthalten, bestimmte Agenten ähnlich UbiWise selbst zu steuern, um Rückschlüsse auf die Funktionalität einer simulierten Anwendung ziehen zu können. Es wird keine Zwischenstufe zwischen dem Agenten und der Umgebung modelliert, obwohl in manchen Beispielszenarien eine solche Infrastruktur vorliegen muss (zum Beispiel die Wireless LAN Stärke). Vielmehr wird dies auf höherer Ebene abgebildet, indem Orte Kontextinformationen bieten und

die Nutzer direkt mit den Werten umgehen können. Auf dieser Ebene ist auch die Kommunikation zwischen den Komponenten, so solche in dem Szenario benötigt wird, auf den unteren Leveln irrelevant, da ihre Existenz vorausgesetzt wird.

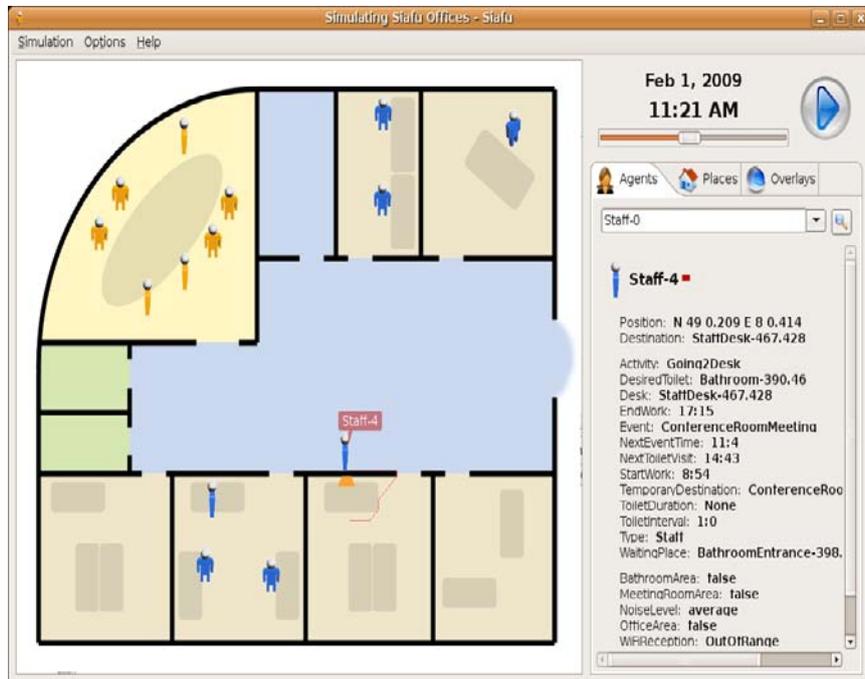


Abbildung 7. Fertiges Szenario in Sifafu, stellt eine Büroumgebung dar

Die Benutzung des Simulators erfolgt klar strukturiert, Verfeinerungen aus den bestehenden Modellen können durch die objektorientierte Strukturierung leicht schrittweise implementiert werden, so dass ein Nutzer schnell und – auch durch die Erstellung der Modelle durch Karten – bequem Ergebnisse bekommen kann. Jedoch ist durch starke Fixierung des Simulators auf eine Welt, in der Agenten agieren, kaum Flexibilität vorhanden, da sich ein System nicht auf verschiedenen Ebenen des Ubiquitous Computing beschreiben lässt. Dadurch ist, auch wenn die Entwickler des Simulators diesen als Kontextsimulator anbieten, aus Tabelle 1 nur *User Context* und *Physical Context* praktikabel simulierbar.

Aithne Project Simulator & Emulator Das Aithne-Projekt befasste sich mit Simulations- und Emulationstestbeds, woraus die in Artikel [3] beschriebene Simulator- und Emulatorumgebung entstanden ist. Ähnlich vieler Simulatoren für Sensornetze nutzt er einen etablierten Netzwerksimulator, in diesem Fall ns-2, für die Simulation der (auch drahtlosen) Kommunikation. Ähnlich plattformspezifischer Simulatoren wie Tossim kann der Entwickler dank der Emulation die Simulationsprogramme direkt für die Plattformen schreiben, auf denen die Anwendungen laufen sollen, vorausgesetzt, es existiert bereits eine Middle-

ware hierfür oder der Entwickler hat die Zeit zu deren Entwicklung (siehe hierzu Abbildung 8).

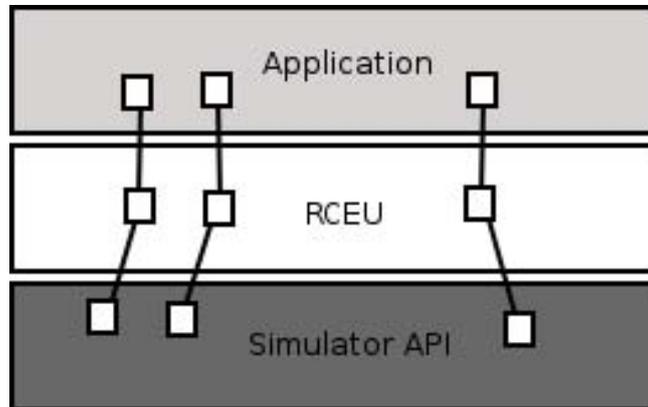


Abbildung 8. Schema der Emulation: Die *Replaceable Code Emulation Unit* bildet die Verbindung zwischen der API der Applikation und jener des Simulators

Wie vorher bereits beschrieben in 2.2 wurden in einer Analyse des Artikels zum Aithne Simulator die vier Abstraktionen Umgebung, Sensor, Aktor und Applikation als Grundbausteine eines ubiquitären Systemes ausgemacht. Zur Modellierung der Umgebung wird ein Rasterbasierter Ansatz verwendet, da die Haupteigenschaft einer Umgebung und der in ihr enthaltenen Objekte der Ort ist.

Die Objekte sind entweder Sensoren oder Aktoren. Sensoren werden zur Erhaltung der Flexibilität nur durch Grundeigenschaften beschrieben. Diese Charakteristiken sind:

- Aktivität oder Passivität des Sensors
- interne oder externe Messung
- periodisches oder sporadisches Auftreten eines Ereignisses

Da Sensoren, auch desselben Types, starken Abweichungen in Genauigkeit und sonstigen Eigenschaften unterliegen, wird eine Sensorpipeline zur Modellierung dieser Eigenschaften genutzt. So können aus einer Ebene erhaltene Werte blockiert oder verändert werden. Messungen werden durch Aufruf der von Ebenen implementierten Funktion *retrieve()* ausgeführt. Aktive Sensoren messen Eigenschaften, während passive Sensoren durch Änderungen in der Umgebung aktiviert werden.

Aktoren besitzen die gleichen Grundeigenschaften wie Sensoren, auch wird eine Filterpipeline verwendet, jedoch in die entgegengesetzte Richtung. Anstatt jedoch Werte durch auslesen zu „messen“, tritt für eine Aktor ein Ereignis auf,

welches nun beispielsweise die Lautstärke um den Ort des Aktors herum in einer Lautstärkeebene aktualisiert.

Durch Ebenen werden mehrere Aspekte der Umgebung dargestellt, dies kann einerseits die Topologie sein, jedoch auch Lichtstärke, Lautstärke oder Temperatur an einem bestimmten Ort. Jede Ebene ist normalerweise unabhängig von den anderen, es können vom Benutzer des Simulators jedoch auch Abhängigkeiten unter mehreren Ebenen definiert werden. Ein Beispiel für einen solchen Ebenenstapel zeigt Abbildung 9. Prinzipiell ist eine Ebene ein zweidimensionales Raster einer festen Größe, die Granularität der Ebene kann individuell angepasst werden (z.B. ein feines Raster für RFID-Leser, jedoch ein grobes Raster für die Temperatur).

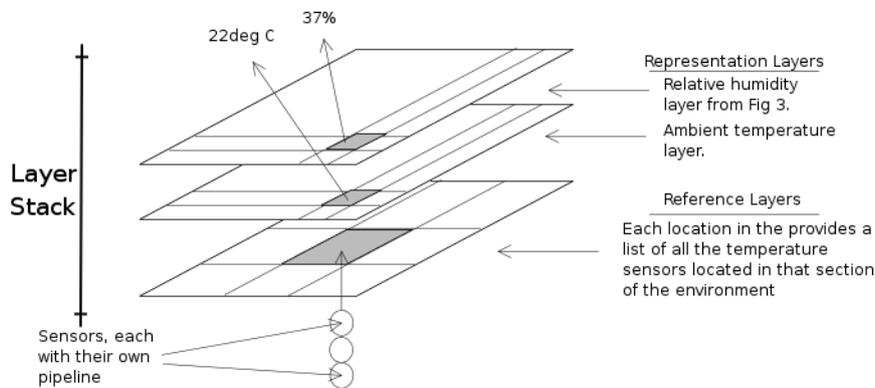


Abbildung 9. Ein Ebenenstapel bestehend aus zwei Repräsentations- und einer Referenzebene

Es gibt zwei verschiedene Arten von Ebenen:

Representation Layers werden benutzt, um die Eigenschaften der physikalischen Umgebung zu modellieren, also zum Beispiel Topologie oder Feuchtigkeit. Es muss die Methode *populate*, welche die Eigenschaftrepräsentation instantiiert, und die Methode *transform* implementiert werden, welche zur Änderung des Status der Ebene verwendet wird.

Durch *Reference Layers* wird die Organisation und die Ablage von bereits erstellten Komponenten wie Sensoren und Aktuatoren ermöglicht. Eine Verbindung zwischen Komponenten und Ebenen kann erstellt werden, so daß ein Nutzer eine Repräsentationsebene mit einer Referenzebene verbinden kann, so dies benötigt wird (z.B. sollte eine Referenzebene für Thermosensoren mit einer Temperatur-Repräsentationsebene verbunden werden). Auch zwischen Repräsentationsebenen ist eine Kopplung möglich durch die Abfrage der Werte einer anderen Ebene in der Implementierung der *transform*-Methode der abhängigen Ebene.

Kommunikation zwischen Knoten wird direkt über den Netzwerksimulator geleitet, dies wird dadurch ermöglicht, daß alle Netzwerkaktivität durch die *Replaceable Code Emulation Unit* (RCEU) weitergereicht wird, wobei durch Implementierung der Middleware Netzwerkereignisse gekennzeichnet sind. Desweiteren gibt es einen *Location Manager*, welcher für mobile Geräte die Ebenenaktualisierung vornimmt, und einen Logmanager, der dem Nutzer die Simulationsausgabe liefert. Diese Architektur wird in Abbildung 10 zusammengefasst.

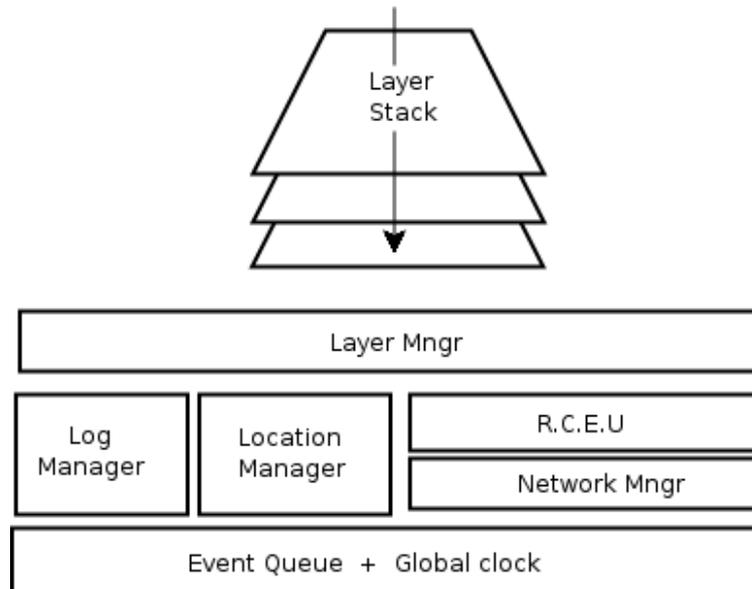


Abbildung 10. Schema der Simulatorarchitektur mit den Schlüsselkomponenten

Durch den stark generalisierten Aufbau des Simulators, entstanden durch sorgfältige Analyse der Erfordernisse, ist eine hohe Flexibilität erreicht worden. Gleichzeitig wird jedoch durch die Architektur die Möglichkeit geschaffen, auf der verschiedene Verfeinerungsebenen möglich sind. Ein Vorteil dieses Simulators ist auch die Emulation, denn hierdurch entfällt eine doppelte Entwicklung von Applikationen für ein zu simulierendes System, gerade dies jedoch bringt für einen Entwickler, der sich entweder noch nicht für eine Plattform entschieden hat oder aber die Simulation nur zum schnellen Ausprobieren nutzen möchte, einen unangenehmen Mehraufwand mit sich.

4 Zusammenfassung

Im folgenden sollen ein übersichtlicherer Vergleich der Eignung der Simulatoren für verschiedene, für Ubiquitous Computing wichtige Eigenschaften aufgeführt werden. Hierzu sollen die Umgebung, Mobilität, Netzwerkkommunikation, Hard-

wareigenschaften wie beispielsweise Batterieentladung und Applikationen betrachtet werden, da diese, auch wenn sie teilweise exemplarisch sind, Eignungsbereiche und -grenzen der beschriebenen Simulatoren aufzeigen.

4.1 Vergleich der vorgestellten Simulatoren

Eine Abbildung von Umgebungen ist mit VisualSense in topologischer Hinsicht in Form von Hindernissen verfügbar, in anderen physikalischen Merkmalen jedoch nur über Aktoren und entsprechende Sensoren realisierbar, wodurch dies sehr umständlich zu modellieren ist. In PythonDEVS (stellvertretend für den DEVS-Formalismus) lässt sich eine Umgebung beliebig feingranuliert modellieren, hierin gibt es jedoch keine Unterstützung in Form von vordefinierten Modellen, so daß auch hier relativ hoher Aufwand betrieben werden muss, um eine Umgebungsmodellierung zu erhalten. Sifafu hingegen bietet durch eine Modellierung über Karten und Overlays eine leicht zu erstellende und dennoch sehr generelle Abbildung von Umgebungen. Der Simulator des Aithne-Projektes bietet einen ähnlichen Ansatz durch Modellierung von Umgebung durch mit anderen Komponenten verbindbare Rasterebenen.

Mobilität ist in VisualSense durch vordefinierte Modelle leicht zu simulieren, dies ist auch in PythonDEVS zu bewerkstelligen, wenn auch mit erheblich höherem Aufwand. Durch das agentenbasierte Design Sifafus ist ein Mobilitätsmodell fast automatisch enthalten, ebenso eine Mobilitätssimulation. Der Simulator des Aithne-Projektes stellt ein Mobilitätsmodell durch Translation auf einer mit der Komponente verbundenen Ebene zur Verfügung.

Ereignisse aus dem Bereich des *Computing Contexts* aus Tabelle 1, wie beispielsweise ein Batteriemodell, lassen sich aufgrund der weiter oben beschriebenen Herkunft von VisualSense leicht modellieren. In PythonDEVS ist dies wiederum mit eigener Modellbildung verbunden, wobei man hier je nach Verfeinerungsgrad mit höherem Aufwand rechnen muß. Ein einfaches Batteriemodell lässt sich aber leicht implementieren, wie [4] zeigt. In Sifafu ließe sich durch Nutzung der Statusvariablen und entsprechenden Anpassungen des Agentenmodelles ein Modell für manche Bereiche des Computing Contexts erstellen, jedoch ist hierfür ein hoher Aufwand vonnöten. Im Aithnesimulator selbst ist für solche Ereignisse keine Modellierung möglich, jedoch kann die Middleware diese Aufgabe plattformspezifisch übernehmen.

Netzwerkkommunikation ist in VisualSense durch die allgemein zur Kommunikation zwischen Komponenten einsetzbaren Ports und die Erweiterung dieser zum Channel leicht modellierbar, auch mit beliebigem Störmodell. PythonDEVS bietet nur Ports zur Kommunikation zwischen Komponenten an, ähnliche Konstrukte wie Kanäle oder Störmodelle sind jedoch auch hier leicht zu implementieren. Sifafu bietet keine Konstrukte für Netzwerkkommunikation an, vielmehr können Agenten direkt auf die Umgebungswerte zugreifen, es fehlt also ein Modell der In-

frastrukturschicht, wodurch auch keine Netzwerkkommunikation benötigt wird. Einen ganz anderen Weg geht hier der Aithne Simulator, dieser fängt über die Plattformmiddleware die Netzwerkaufrufe ab und lässt diese durch den Netzwerksimulator ns-2 simulieren.

Applikationen werden in VisualSense über die Komponenten beziehungsweise Zusammenschlüsse solcher realisiert. Diese lassen sich nicht nur durch Blockschaltbilder beschreiben, sondern auch in Java implementieren. Ebenso hier wie auch bei PythonDEVS und Siafu gibt es keine klare Trennung zwischen modellierten Komponenten und der jeweiligen Applikation. Der Aithne Simulator bietet ein Applikationsinterface über die Emulationseinheit an, so dass die Applikationsimplementierung unabhängig von dem Simulationssetting ist.

4.2 Fazit

Alle hier vorgestellten Simulatoren berücksichtigen einen Teil der Erfordernisse für die Simulation von Ubiquitären Systemen und Applikationen, jedoch lässt sich mit vertretbarem Aufwand in keinem Fall das gesamte Feld abdecken.

VisualSense beispielsweise stützt sich auf die kleinstbauteilige Beschreibung von Komponenten, wodurch sowohl Gesamtapplikationen nur mit hohem Aufwand aufgebaut werden können als auch der Umgebungsbezug schwer herzustellen ist. Der Simulator des Aithne Projektes ist zwar sehr flexibel, was Komponenten und Umgebung angeht, jedoch wird er nur dann in Frage kommen, wenn die Entscheidung zur Entwicklung eines Systemes bereits gefallen ist.

Somit genügt es nicht, mit einem Simulator alle Schritte durchzuführen, sondern es müssen, je nach Projektstatus und Art des Projektes verschiedene Werkzeuge gewählt werden, beispielsweise kann für eine grobe Simulation einer Applikation DEVS verwendet werden, zur Erprobung dieser Applikation in einer simulierten Stadt Siafu und zur endgültigen Entwicklung mit gleichzeitiger Testmöglichkeit der Simulator des Aithneprojektes.

Literatur

1. Weiser, M. 1993. Some computer science issues in ubiquitous computing. *Commun. ACM* 36, 7 (Jul. 1993), 75-84. DOI= <http://doi.acm.org/10.1145/159544.159617>
2. Weiser, M. and Brown, J. S. 1997. The coming age of calm technology. In *Beyond Calculation: the Next Fifty Years*, P. J. Denning and R. M. Metcalfe, Eds. Copernicus, New York, NY, 75-85.
3. Reynolds, V., Cahill, V., and Senart, A. 2006. Requirements for an ubiquitous computing simulation and emulation environment. In *Proceedings of the First international Conference on integrated internet Ad Hoc and Sensor Networks (Nice, France, May 30 - 31, 2006)*. *InterSense '06*, vol. 138. ACM, New York, NY, 1.

4. Antoine-Santoni, T., Santucci, J. F., De Gentili, E., and Costa, B. 2007. Modelling & simulation oriented components of wireless sensor network using DEVS formalism. In Proceedings of the 2007 Spring Simulation Multiconference - Volume 2 (Norfolk, Virginia, March 25 - 29, 2007). Spring Simulation Multiconference. Society for Computer Simulation International, San Diego, CA, 299-306.
5. Martin, Miquel; Nurmi, Petteri, A Generic Large Scale Simulator for Ubiquitous Computing, Mobile and Ubiquitous Systems: Networking & Services, 2006 Third Annual International Conference on , vol., no., pp.1-3, July 2006
6. Baldwin, P.; Kohli, S.; Lee, E.A.; Xiaojun Liu; Yang Zhao, Modeling of sensor nets in Ptolemy II, Information Processing in Sensor Networks, 2004. IPSN 2004. Third International Symposium on , vol., no., pp. 359-368, 26-27 April 2004
7. Hong, D., Chiu, D. K., and Shen, V. Y. 2005. Requirements elicitation for the design of context-aware applications in a ubiquitous environment. In Proceedings of the 7th international Conference on Electronic Commerce (Xi'an, China, August 15 - 17, 2005). ICEC '05, vol. 113. ACM, New York, NY, 590-596.
8. J.J. Barton, and V. Vijayaraghavan : "UBIWISE, A Simulator for Ubiquitous Computing Systems Design", Technical Report HPL-2003-93, HP Laboratories, Palo Alto, 2003.
9. Markus Bylund and Fredrik Espinoza. Using quake III arena to simulate sensors and actuators when evaluating and testing mobile services. In CHI 2001 Extended Abstracts, pages 241-242. ACM, March-April 2001. Short Talk.
10. Köpke, A., Swigulski, M., Wessel, K., Willkomm, D., Haneveld, P. T., Parker, T. E., Visser, O. W., Lichte, H. S., and Valentin, S. 2008. Simulating wireless and mobile networks in OMNeT++ the MiXiM vision. In Proceedings of the 1st international Conference on Simulation Tools and Techniques For Communications, Networks and Systems & Workshops (Marseille, France, March 03 - 07, 2008). ICST (Institute for Computer Sciences Social-Informatics and Telecommunications Engineering), ICST, Brussels, Belgium, 1-8.
11. Sundresh, S., Kim, W., and Agha, G. 2004. SENS: A Sensor, Environment and Network Simulator. In Proceedings of the 37th Annual Symposium on Simulation (April 18 - 22, 2004). Annual Simulation Symposium. IEEE Computer Society, Washington, DC, 221.
12. <http://ptolemy.eecs.berkeley.edu/>, am 28.01.2008

Formale Modelle für Ubiquitäre Informationstechnologien

David Lisowski

Telecooperation Office, Institut für Telematik, Universität Karlsruhe (TH)
Betreuer: Till Riedel

1 Einleitung

Der Begriff *Ubiquitous Computing* (Synonyme: *Pervasive Computing*, zu deutsch *Ubiquitäre Informationstechnologien*, kurz: *UbiComp*) wurde 1988 von Mark Weiser eingeführt und in seinem Artikel „The Computer for the 21st Century“ (siehe [UbiCWei]) geprägt. Ubiquitär bedeutet dabei allgegenwärtig oder überall verbreitet.

Weisers Vision, dass die Computer so allgegenwärtig sind, dass sie als solche nicht mehr wahrgenommen werden, hat sich bewahrheitet. Personal Computer verschwinden und werden durch intelligente Gegenstände, welche den Menschen im Alltag behilflich sein sollen, ersetzt. Ubiquitäre Informationstechnologien bieten dabei Berechnungs- und Kommunikationsdienste an, die transparent für den Anwender und immer und überall verfügbar sind. Ubiquitous Computing stellt die dritte Ära der Computergeschichte dar, in welcher sich Computer nahtlos und vernetzt in die Umwelt einbetten, um jeder Person zur Verfügung zu stehen. Die erste Ära bildet das Mainframe Computing, mit nur einem Großrechner für mehrere Anwender (meistens Experten und Wissenschaftler). Die verbleibende zweite Ära betrifft die meisten von uns, denn es handelt sich dabei um das Personal Computing mit üblicherweise einem Computer pro Anwender. Dieser erfordert zwar auch ein gewisses Vorwissen, aber man muss kein Experte mehr sein.

Im Ubiquitous Computing verfolgt man den Ansatz, bestehende Basistechnologien zu kombinieren, anstatt neue zu erfinden. Solche Basistechnologien sind z. B. Sensoren/Aktoren, Auto-ID Systeme, Positionierungssysteme und Drahtlosnetzwerke. Genauer kann man unter [UbiCompEn] nachlesen.

Ubiquitous Computing begegnet uns im Alltag u.a. in der Logistik, Supply-Chain-Management, SmartHome und SmartOffice, Augmentation, Wissensmanagement, Sicherheits- und Notfallmanagement, Medizin und Entertainment.

Ubiquitäre Informationstechnologien kommunizieren über ein mobiles Ad-Hoc-Netz (siehe [AdHocWiki], [AdHocLex] und [VerSysWiki]).

Ubiquitous Computing und Formale Modelle Das Gebiet ubiquitärer Informationssysteme hat sich sehr rasch entwickelt und viele interessante Neuerungen in den letzten Jahren hervorgebracht. Leider mangelt es aber an weit fortgeschrittenen und akzeptierten formalen Modellen zur Beschreibung solcher

Systeme. Gründe dafür sind Schwierigkeiten wie ständige Zustandsänderungen von Geräten, die Unzuverlässigkeit von Ad-hoc-Netzen, Autonomie der Komponenten und vor allem der Einsatzort, der ebenfalls modelliert werden muss.

Ebenfalls berücksichtigt werden müssen zwei ganz wichtige UbiComp Konzepte: Der Kontext [UCKon] und die Mobilität [Mob], bei der zwischen der physikalischen Mobilität und der logischen unterschieden wird.

Zwei solche Modelle für ubiquitäre Informationstechnologien sind Active Spaces und mobil-synchronisierende Petri-Netze. Zunächst werde ich über die Active Spaces und deren Grundlagen berichten, danach gehe ich auf die mobil-synchronisierenden Petri-Netze, u. a. auch auf Petri-Netze und die Erweiterung der mobil-synchronisierenden Petri-Netze, ein.

2 Active Spaces

2.1 Einleitung

Es gibt bereits verschiedene Modelle, um das Verhalten und die Eigenschaften von klassischen Systemen mit starrer Vernetzung zu spezifizieren. Dazu gehören unter anderem Automaten, Pi Calculus und Prozess Algebra. Doch geht es explizit um ubiquitäre Informationssysteme, wird es aufgrund der in der Einleitung beschriebenen Probleme schwieriger.

In diesem Kapitel gehe ich auf ein formales Modell ein, das auf dem Ambient Calculus und der Ambient Logic basiert: die sogenannten Active Spaces [AcSpPap].

Als erstes beschreibe ich den Ambient Calculus. Dieser ist wichtig zum Modellieren verschiedener Arten von Ressourcen (Geräte, Anwender, Services...), zum Spezifizieren von Operationen, die ausgeführt werden können und von Abläufen, die in einem solchen System auftreten können.

Als nächstes gehe ich auf Ambient Logic ein, welche geeignet ist für das Beschreiben und Verifizieren gewisser Eigenschaften von ubiquitären Informationssystemen.

Zu guter Letzt widme ich mich ganz den Active Spaces.

2.2 Ambient Calculus

Der Ambient Calculus wurde von Luca Cardelli und Andrew Gordon entworfen, um die Bewegungsabläufe (Ein- und Ausgabe-Operationen) von Prozessen und Geräten zu beschreiben. Als Ambient wird dabei ein begrenzter Platz bezeichnet, in dem Berechnungen durchgeführt werden. Die Grenze spielt dabei eine wichtige Rolle, denn sie bestimmt, was innerhalb bzw. außerhalb eines Ambients stattfindet. Beispiele für solche Ambients sind eine Webseite (begrenzt durch eine Datei) oder auch ein virtueller Adressraum (begrenzt durch die Reichweite der Adresse).

Ambients können auch untereinander verschachtelt werden, so dass sie eine Baumhierarchie bilden. Eine solche Verschachtelung tritt beispielsweise beim

Einbinden eines Java-Programmes in einer Webseite auf, die ihrerseits in einem Browser dargestellt wird. Dieser Browser wird auf einem Betriebssystem ausgeführt, welches auf einem Notebook installiert wurde. Dieser Sachverhalt ist in Abbildung 1 dargestellt. Natürlich ist das ein simples Beispiel. In der Realität sieht das dann etwas komplexer aus. Es kann von jedem Ambient mehrere Exemplare geben, sprich mehrere Browser oder natürlich auch mehrere Webseiten usw.

Außer der Verschachtelung, haben Ambients noch andere Eigenschaften: Jeder Ambient hat einen Namen und kann als Ganzes durch Hilfe von Ein- und Ausgabe-Operationen in einen anderen Ambient bzw. aus einem anderen Ambient heraus verschoben werden. Der Ambient Calculus ist sehr ausdrucksstark und eignet sich für viele gewünschte Eigenschaften in UbiComp und bietet viele primitive Operationen an, auf die ich nun etwas genauer eingehen will.

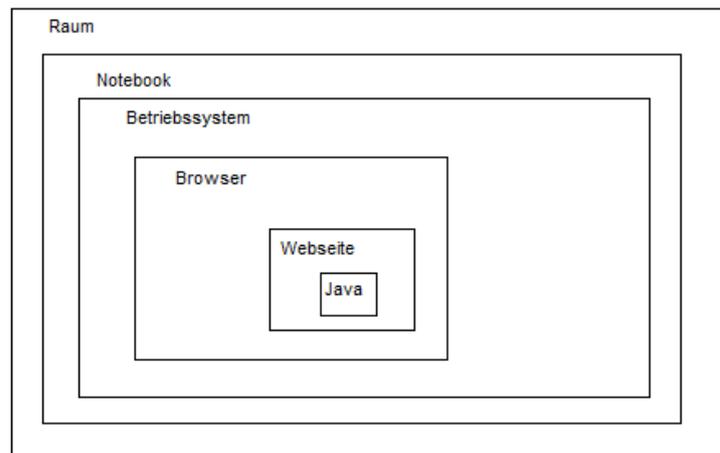


Abbildung 1. Verschachtelung von Ambients

Primitive Operationen Im Ambient Calculus werden alle Einheiten als Prozesse $n[P]$ beschrieben, wobei n der Name des Ambients ist und P den Prozess bezeichnet, der innerhalb der Grenzen des Ambients läuft.

0 steht dabei für den inaktiven Prozess. Mit Hilfe des Operators $|$ kann das parallele Ausführen (Komposition) von Prozessen beschrieben werden. $!P$ steht für die unbegrenzte Kopie des Prozesses P . $!P$ kann so viele parallele Kopien wie benötigt von P erstellen und ist damit äquivalent zu $P \mid !P$. Die schon weiter

oben beschriebene Verschachtelung von Ambients wird durch $n[P_1 \mid \dots \mid P_j \mid m_1[\dots] \mid \dots \mid m_k[\dots]]$ dargestellt, wobei j für die Anzahl der Prozesse mit den Namen P_1, \dots, P_j und k für die Ambients mit den Namen m_1, \dots, m_k steht.

Es gibt ebenfalls Prozesse, die durch Hilfe bestimmter Fähigkeiten den Zustand um sich herum ändern können. Diese Fähigkeiten sind:

1. Das Betreten eines Ambients mittels *in*.
2. Das Verlassen eines Ambients mittels *exit*.
3. Das Öffnen eines Ambients mittels *open*.

Anhand von Abbildung 2 möchte ich noch genauer auf diese drei Fähigkeiten eingehen.

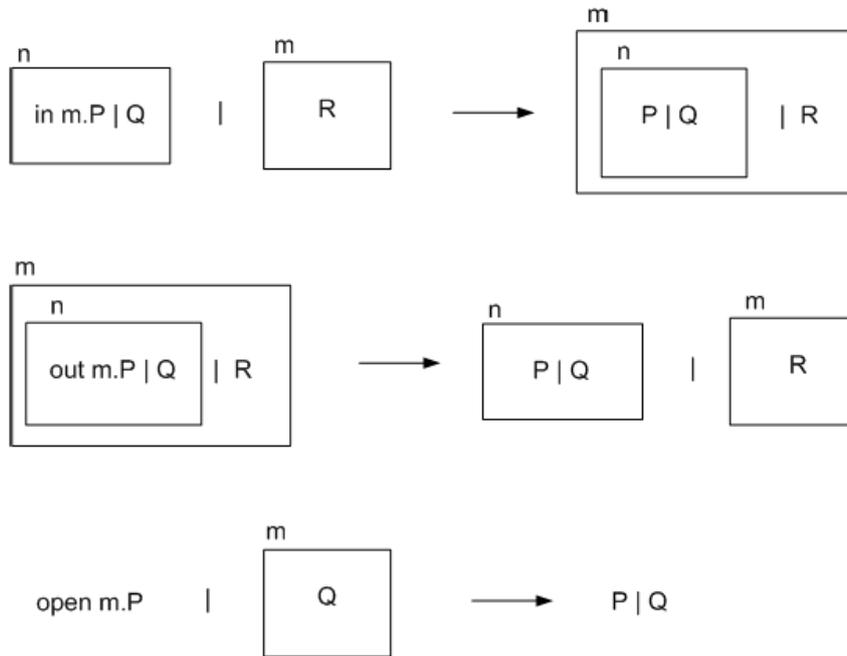


Abbildung 2. spezielle Fähigkeiten

Wir haben Ambients mit den Namen n und m und die Prozesse P, Q, R . Im ersten Beispiel befindet sich im Ambient n der Prozess $in\ m.P \mid Q$, was nichts anderes heißt, dass n m betreten soll. Gibt es allerdings keinen solchen Ambient m , wird der Befehl solange blockiert, bis so ein m existiert. Für den Fall, dass es mehrere Ambients m gibt, darf man einen beliebigen aussuchen. In unserem Beispiel ist die Auswahl allerdings eindeutig, da es nur einen einzigen Ambient m mit Prozess R gibt. Das Endresultat ist ein Ambient m (unverändert mit dem Prozess R), der den Ambient n mit dem Prozess $P \mid Q$ beinhaltet. Diese Umformung lässt sich durch eine Reduktionsregel ausdrücken als:

$$n[in\ m.P \mid Q] \mid m[R] \rightarrow m[n[P \mid Q] \mid R]$$

Das zweite Beispiel läuft nach dem selben Muster ab, nur umgekehrt. Im Ambient m befindet sich der Prozess R und der Ambient n mit dem Prozess $out\ m.P \mid Q$. Dieser Prozess sagt aus, dass der umliegende Ambient n den Ambient m verlassen soll. Das Ergebnis sind dann n mit $P \mid Q$ und m mit R . Sollte es keinen Eltern-Ambient m geben, wird der Befehl solange blockiert, bis es einen gibt.

Formal sieht das dann so aus: $m[n[out\ m.P \mid Q] \mid R] \rightarrow n[P \mid Q] \mid m[R]$.

Mit Hilfe von $open\ m.P$ ist es möglich die Grenze eines Ambients m , der sich auf der gleichen Ebene wie $open$ befindet, aufzulösen. Gibt es keinen Ambient m , wird wie bei den vorangegangenen Beispielen solange der Befehl blockiert, bis es so einen Ambient gibt. Formal: $open\ m.P \mid m[Q] \rightarrow P \mid Q$.

Eine Ausgabe-Aktivität gibt eine Fähigkeit in der Form von $\langle M \rangle$ in den lokalen Äther des umgebenen Ambients ab. Eine Eingabe-Aktivität $(x).P$ hingegen fängt eine Fähigkeit vom lokalen Äther ab und verknüpft es zu einer Variablen x innerhalb des Bereichs.

Der Abgrenzungs-Operator $(vn)P$ erzeugt innerhalb des Bereichs vom Prozess P einen einzigartigen Namen, welcher dazu dient Ambients zu benennen und mit Ambients durch Hilfe des Namens zusammen zu arbeiten.

2.3 Ambient Logic

Viele Eigenschaften ubiquitärer Informationssysteme, wie z. B. Services, die überall angeboten werden können und Mobilität, basieren auf dem Konzept des Einsatzortes. Zum Beschreiben und Verifizieren dieser Eigenschaften ist die Ambient Logic, welche zur Familie der modalen Logik gehört, sehr gut geeignet. Dabei können in logischen Formeln der Ambient Logic folgende Ausdrücke auftauchen:

1. aussagenlogische Ausdrücke wie *wahr*, Negation, Disjunktion
2. Operatoren für räumliche Abfolgen wie Komposition, Ort
3. Operatoren für zeitliche Abfolgen wie möglicherweise, notwendig
4. Quantoren (Existenz- und Allquantor)

Die Erfüllbarkeit $P \models A$ besagt, dass der Prozess P die geschlossene Formel A , d. h. alle Variablen sind gebunden und nicht frei, erfüllt.

Für die zeitlichen Abfolge bedeutet $P \rightarrow P'$, dass der Ambient P durch die Anwendung einer Reduktionsregel zu P' reduziert werden kann. Durch $P \rightarrow^* P'$ wird die reflexiv, transitive Hülle dieser Reduktionsbeziehung beschrieben.

Analog dazu steht $P \downarrow P'$ dafür, dass sich der Ambient P und der Ambient P' räumlich gesehen innerhalb der gleichen Verschachtelungsebene befinden, wobei $P \downarrow^* P'$ die reflexiv, transitive Hülle dieser Beziehung ist.

2.4 Active Spaces

Active Spaces sind ein formales Modell basierend auf Ambient Calculus und Ambient Logic. Damit sind Aussagen und Beweise über einige Eigenschaften

wie Services, die überall irgendwie vorhanden sind und Mobilität von Geräten und Programmen möglich.

Ein Active Space, sowie die in ihm enthaltenen Ressourcen und dessen Kontext werden dabei als Ambients im Ambient Calculus modelliert. Dabei umfassen die Ressourcen Software-Komponenten, Geräte, Anwender, physikalische Objekte (z. B. Stuhl, Tisch) und andere Active Spaces. Software-Komponenten sind dabei verschiedene (miteinander kommunizierende) Services (z. B. Authentifizierung) und Programme (z. B. Musik-Player). Geräte wie Sensoren, PDAs und Laptops sind Hardware-Einheiten in einem Active Space.

Active Spaces können ebenfalls andere Active Spaces beinhalten. Aufgrund der Tatsache, dass alle Ressourcen als Ambients modelliert werden, sind nun räumliche Beziehungen zwischen den verschiedenen Ressourcen baumartig darstellbar. Ein Gerät besteht beispielsweise aus einer oder mehreren Software-Komponenten und ein Anwender umfasst ein oder mehrere Geräte. Ein konkreteres formales Beispiel für ein Gerät dev mit zwei Dateien ($file1$ und $file2$), vier Software-Komponenten, wovon zwei Programme ($app1$ und $app2$) und zwei Services ($serv1$ und $serv2$) sind wäre:

$$dev[file1 \mid file2 \mid app1[P] \mid app2[Q] \mid serv1[R] \mid serv2[S] \mid Z]$$

Dabei stehen P, Q, R, S, Z für Prozesse. P läuft innerhalb des Ambients $app1$ und umfasst dessen Inhalt. Analog dazu Q, R und S . Z ist der Vollständigkeit halber aufgeführt und umfasst alle anderen Inhalte von dev und kann auch die Komposition anderer Prozesse sein, welche wiederum Ambients sein können.

Der Kontext wird durch einen Prozess im Ambient Calculus modelliert und er kann dadurch mit einem bestimmten Ambient verbunden werden. Ein Anwender kann mehrere persönliche Geräte besitzen und sich in einem gewissen Kontext befinden. Formal: $user1[dev1[P1] \mid dev2[P2] \mid cxt1]$

$dev1$ und $dev2$ werden dabei wie oben beschrieben modelliert, $P1$ und $P2$ können Kompositionen verschiedener Prozesse sein, die Dateien, Software-Komponenten etc. darstellen. $cxt1$ kann auch Komposition von Kontexten sein, also kann $cxt1$ beispielsweise Ort des Anwenders, seine Verfügbarkeit beim Chatten und dessen Position enthalten. Für einen Anwender der im *Raum 42* sitzt und zum Chatten bereit ist ließe sich der Kontext $cxt1$ wie folgt formalisieren:

$$cxt1 \cong Raum42 \mid verfuegbar \mid sitzend$$

Ein Active Space kann also viele Geräte und Anwender enthalten und sich in einem gewissen Kontext befinden. Ein umfangreicheres Beispiel für einen Active Space ist in Abbildung 3 dargestellt. Die Struktur lässt uns dabei genaue Aussagen über verschiedenen Beziehungen zwischen Geräten, Anwendern etc. machen. Geräte, die sich direkt innerhalb eines Active Spaces befinden, sind öffentlich ($dev3$ und $dev4$), während solche, die sich innerhalb des Active Spaces eines Anwenders befinden persönlich sind, d. h. nur für diesen Anwender bestimmt sind ($user5$ mit Gerät $dev5$ in der Abbildung 3). Das Gleiche gilt natürlich auch für Software-Komponenten. Allerdings besteht die Möglichkeit, dass ein Benutzer anderen Anwendern die Erlaubnis gibt, seine persönlichen Dinge zu benutzen.

Ein Active Space kann aber nicht nur Ambients, sondern auch andere Active Spaces beinhalten.

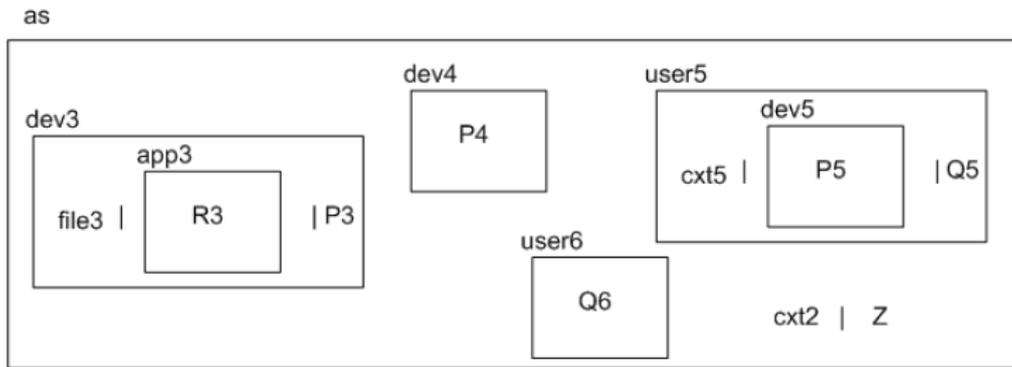


Abbildung 3. Beispiel für einen Active Space

Verschiedene Prozesse haben verschiedene Fähigkeiten. Zum Beispiel kann ein Gerät einige Komponenten hosten, aber andere nicht. Auf einem PDA *pda1* kann zum Beispiel eine Komponente namens *slideControlComponent* laufen, die die Kontrolle von Folien einer Präsentation erlaubt, aber eine Komponente namens *faceRecognizer*, die der Gesichtserkennung dient, läuft auf dem selben PDA nicht. Formal sieht das dann so aus:

$$pda1 \models canHost(slideControlComponent)$$

$$pda1 \not\models canHost(faceRecognizer)$$

Operationen in einem Active Space Alle Operationen werden als Ambients modelliert. Man unterscheidet zwischen folgenden Operationen:

1. *start in dev.P* steht für den Startvorgang einer Komponente *P* auf einem Gerät *dev*.
2. *stop in dev.P* steht für das Stoppen einer Komponente *P*, die auf einem Gerät *dev* läuft.
3. *enterDev in as.dev* bedeutet, dass das Gerät *dev* den Active Space *as* betritt.
4. *exitDev out as.dev* heißt, dass das Gerät *dev* den Active Space *as* verlässt.
5. *enterUser in as.user* bedeutet, dass der Anwender *user* den Active Space *as* betritt.
6. *exitUser in as.user* heißt, dass der Anwender *user* den Active Space *as* verlässt.
7. *moveDev out as.user* heißt, dass der Anwender *user* ein Gerät für andere zur Verfügung stellt, d.h. das Gerät wird aus dem Anwender Ambient raus in den umliegenden Active Space Ambient verschoben.
8. *send in a.⟨E⟩*: Ereignis an einen Ambient *a* senden. Ereignisse werden durch eine asynchrone Ausgabe-Aktivität dargestellt und können durch Prozesse ausgelöst werden. Jedes Ereignis hat dabei einen gewissen Anwendungsbereich, welcher dem Ambient entspricht, in dem sich das Ereignis befindet. Nur andere Prozesse in diesem Ambient können das Ereignis abfangen.

9. *changeContext in a.[remove[cxt1] | add[cxt2]]* steht für das Ändern des Kontextes von Ambients (hier: *cxt1* wird entfernt und *cxt* kommt dazu).

Grundlegende Entwurfsideen Alle Einheiten in einem Active Space sind durch eine Baumhierarchie mit dazugehörigen Grenzen darstellbar. Mobilität von Objekten entspricht einer Grenzüberschreitung des entsprechenden Objektes und Sicherheit ist dabei die Fähigkeit bzw. das Unvermögen Grenzen zu überqueren. Das Zusammenspiel von Prozessen ist durch den dazugehörigen Ambient begrenzt.

Aktivitäts-Modell Eine Aktivität, die als Ambient modelliert wird, besteht aus mehreren Aufgaben (Tasks), welche als Komposition von Prozessen modelliert werden. Zum Beispiel eine Aktivität namens *meeting*, die folgende Aufgaben umfasst: Präsentation zeigen *displayPresentation*, Kontrolle der Präsentation *controlPresentation* und Licht abdunkeln *dimLights*. Formal also:

meeting[displayPresentation | controlPresentation | dimLights]

Eine Aufgabe wird normalerweise so ausgeführt, indem bestimmte Software-Komponenten auf gewissen Geräten laufen. Also sind die Parameter „Komponente“ und „Gerät“ von Bedeutung, um einen Task zu modellieren.

displayPresentation $\cong (c).(d).start\ in\ d.c\ so,\ dass\ c\ |=\ isComponent \wedge canPerform(slideDisplay)\ and\ d\ |=\ isDevice \wedge canHost(c)$

In dem obigen Beispiel haben wir den Task *displayPresentation* und 2 Eingaben *c* und *d*, wobei der Task im Wesentlichen die Komponente *c* im Gerät *d* startet. Dabei wird durch *canPerform(slideDisplay)* gefordert, dass die Komponente *c* in der Lage ist Folien darzustellen, während *d |= canHost* sicherstellt, dass das Gerät *d* die Komponente unterstützt.

Nun erweitern wir unsere Operationen in einem Active Space um eine weitere: Das Ausführen einer Aktivität in einem Active Space wird beschrieben durch *perform in as.activity*. Ob eine bestimmte Aktivität in einem Active Space durchgeführt werden kann, machen wir uns anhand eines Beispiels klar.

performMeeting = $\nabla canPerform(slideControl) \wedge$

$\nabla canPerform(slideDisplay) \wedge (\forall d(as[d] \wedge isLight(d)) \Rightarrow dim(d))$

Um in einem Active Space die Durchführung eines Meetings zu erlauben, muss es irgendeine Komponente innerhalb des Active Spaces geben (∇ -Operator), welche die Darstellung der Folien steuern kann und alle Lichter im Raum müssen verdunkelt sein.

Eigenschaften von Active Spaces

1. Eine bestimmte Aktivität kann in allen Spaces ausgeführt werden, die zu einer bestimmten Menge von Active Spaces gehören.
2. Geräte-Mobilität: Anwender können ihre Geräte in die Umgebung mit einbringen, dies wird modelliert als Bewegung eines Anwenders mit zugehörigem Gerät in einen Active Space, oder formal:

$enterUser$ in $as.user[dev1 \mid P] \mid as[Q] \models \diamond as[\nabla event(dev1)]$ where $\langle dev1 \rangle \models event(dev1)$

3. Kontext-Abhängigkeit: Es kann durchaus passieren, dass es zu Änderungen im Kontext kommt. Active Spaces können sich in einem solchen Fall allerdings neu konfigurieren, wodurch gewisse Fähigkeiten erhalten bleiben. Damit eine Eigenschaft p im Active Space erhalten werden soll, muss wenn $as[cxt1 \mid Q] \models p$ gilt, auch $as[cxt1 \mid Q] \mid changeContext$ in $a.[remove[cxt1] \mid add[cxt2]] \models \diamond p$ gelten.

3 Mobil-synchronisierende Petri-Netze

3.1 Einleitung

Die mobil-synchronisierenden Petri-Netze (MSPN) [MSPNPap], die auf Petri-Netzen basieren, sind ein zweites Modell für ubiquitäre Informationssysteme. Es gibt zwar schon einige Modelle ([MPN], [APN]), die ebenfalls auf Petri-Netzen basieren, die aber nur einige grundlegende Aspekte wie Mobilität von ubiquitären Informationssystemen behandeln. MSPN konzentrieren sich dabei auf andere Eigenschaften wie die Koordination zwischen anbietenden (produzierenden) und fordernden (konsumierenden) Prozessen und Ad-hoc Netze. Geräte und Software-Komponenten werden dabei als spezielle Art von gefärbten Petri-Netzen, die sich an einer bestimmten Lokalität befinden, modelliert.

Diese Lokalität kann durch das Aktivieren bestimmter Übergänge allerdings geändert werden. Übergänge, die Services fordern bzw. anbieten, werden nur durch ihre Existenz schon entdeckt und können nur aktiviert werden, wenn der fordernde und der anbietende Prozess sich am selben Ort befinden.

Lokale Kommunikation wird durch einen Spezialfall der Synchronisation, genauer durch Tokentransfer von einem zum anderen Netz, erreicht.

Natürlich spielt auch die Sicherheit eine Rolle. In einem MSPN wird dies durch die Kenntnis der Ortsnamen und der Service-Namen umgesetzt.

Neben den Petri-Netzen und den MSPN, werde ich noch auf die mobil-synchronisierenden, authentifizierenden Petri-Netze eingehen. Diese ergänzen die Sicherheit um Namenszertifikate, die der Authentifizierung dienen.

3.2 Petri-Netze

Auf Petri-Netze möchte ich nur sehr kurz eingehen. Genaueres unter [PetriWeb].

Ein Petri-Netz ist ebenfalls ein formales Modell zur allgemeinen Beschreibung von Systemen. Es gibt Plätze (Stellen), die als Kreise anschaulich dargestellt werden und Übergänge (Transitionen), die als Rechtecke dargestellt werden, als Knoten, welche durch gerichtete Kanten mit entsprechender Gewichtung verbunden werden. Allerdings existiert keine direkte Verbindung zwischen zwei Plätzen bzw. zwei Übergängen.

Plätze stehen hierbei für Bedingungen und werden mit einem Token (Marke) versehen, wenn die Bedingung erfüllt ist. Die Anzahl der Markierungen ist durch

eine gegebene Kapazität beschränkt. Eine Bedingung kann dabei so ziemlich alles sein, wie z.B. „Der Fernseher läuft.“ oder „Ich bin erkältet“.

Übergänge heißen schaltbereit, wenn die Kosten der Übergänge durch die an den verbundenen Plätzen vorhandenen Marken gedeckt werden können und die Ausgangsstellen noch genügend Kapazitäten haben, um neue Marken aufzunehmen. Zu einem Ereignis gehören immer bestimmte Bedingungen (Vorbedingungen oder Nachbedingungen). Diese Beziehung zwischen Ereignis und Bedingungen wird durch die gerichteten Kanten dargestellt.

Bei einfachen Petri-Netzen können die Marken untereinander nicht unterschieden werden, daher gibt es die farbigen Petri-Netze, die die einfachen Marken um verschiedene Farben erweitern.

3.3 Mobil-synchronisierende Petri-Netze: MSPN

In diesem Abschnitt gehe ich auf ein paar Eigenschaften von MSPN ein und zeige diese dann in einigen Beispielen. Näheres zu dem formalen Unterbau der MSPN findet sich unter [MSPNPap].

Mobil-synchronisierende Petri-Netze sind eine spezielle Art von gefärbten Petri-Netzen mit zwei verschiedenen Farbtypen: Einer für Örtlichkeiten und einer für einfache schwarze Token (nur Plätze werden damit markiert, wie auch schon bei den gewöhnlichen Petri-Netzen). MSPN haben drei Übergangstypen:

1. Autonome/unabhängige Übergänge, welche den einfachen Übergängen der Petri-Netze entsprechen.
2. Synchrone Übergänge werden nur geschaltet, wenn sich am selben Ort ein konjugierter Übergang befindet. D. h. jemand bietet einen Service an (gekennzeichnet durch S!) und jemand anderes fordert diesen (gekennzeichnet durch S?).
3. Bewegungsübergänge mit spezieller Markierung GO entsprechen den autonomen Übergängen, mit der Ausnahme, dass der Ort des Netzes, an dem sie aktiviert werden, verändert werden kann. Deswegen können schwarze Token nur zur Kontrolle eingesetzt werden, aber Orts-Token werden vom Netz benutzt um ein Ziel zu setzen, wenn ein Bewegungsübergang ausgeführt wird.

Jede Verbindung (Kante, Bogen) zwischen einem Übergang und einem Platz ist mit einer Variable gekennzeichnet. Dadurch kann zwischen verschiedenen Orts-Token unterschieden werden. Die spezielle Variable ϵ kennzeichnet dabei Verbindungen zu einem Platz mit einem schwarzen Token.

Jede Variable, die in einem autonomen Übergang auf einer ausgehenden Verbindung auftritt, muss auch auf einer eingehenden Verbindung von diesem Übergang auftauchen, so dass Orts-Token nur von diesem Übergang verbraucht bzw. bewegt werden können, aber nicht in diesem Übergang hergestellt werden können. Für jeden Bewegungsübergang existiert genau ein Platz aus der Vorbedingung, der mit einem Orts-Token gekennzeichnet wurde.

Die Belegung der Plätze mit einem schwarzen Token oder einem Orts-Token heißt Markierung und alle Markierungen bilden zusammen den Zustand des Netzes.

Ein MSPN-System besteht aus mehreren disjunkten MSPN und die Markierung davon ist ein Paar, bestehend aus der Markierung des jeweiligen MSPN und dem dazugehörigen Ort.

Übergänge sind in verschiedenen Modi aktivierbar. Der jeweilige Modus bestimmt, welche Token von der Vorbedingung genommen werden und bildet jede Variable eines Übergangs auf einen Token ab (die Variable ϵ wird auf den schwarzen Token abgebildet usw.).

Kommen wir nun zur Schaltbereitschaft der verschiedenen Übergänge. Bei einem MSPN-System sieht es nach dem Aktivieren eines autonomen Übergangs so aus: Alle Plätze der Vorbedingung haben die selbe Markierung wie vor dem Aktivieren, es fehlen nur die Token, die durch die Aktivierung bewegt wurden, denn diese befinden sich nun in den Plätzen der Nachbedingung. Der Ort ändert sich dabei nicht.

Ein Bewegungsübergang GO verläuft ähnlich, nur der Ort ändert sich dabei in dem jeweiligen aktivierten MSPN des MSPN-Systems. Der neue Ort wird dabei durch den Modus der Verbindung aus der Vorbedingung bestimmt.

Um jetzt die synchronen Übergänge zu beschreiben, müssen wir erstmal festlegen, wann zwei Übergänge kompatibel sind: dies ist genau dann der Fall, wenn einerseits deren Kennzeichnung konjungiert ist (d. h. einer bietet den Service an, während jemand anders ihn fordert) und andererseits jede Variable in den ausgehenden Verbindungen auch in einer eingehenden Verbindung auftritt. Dann gilt bei synchronen Übergängen: zwei kompatible Übergänge werden in einem bestimmten Modus aktiviert und der Zustand des Netzes nach der Aktivierung entspricht dem vor der Aktivierung, nur dass die betroffenen Plätze aus der Vorbedingung ohne die Token sind, die durch die Aktivierung bewegt wurden. Diese Token befinden sich nun auf den Plätzen aus den Nachbedingungen der aktivierten Übergänge.

Ein Beispiel für so ein MSPN ist in Abbildung 4 dargestellt. Der Client würde nun in der in Abbildung 5 dargestellten Reihenfolge mit dem System interagieren. Der erste Prozess befindet sich in einem Ort, der mit den Namen H markiert wurde. Es folgt ein Bewegungsübergang GO, der der Ortsänderung dient, denn der Service kann nur im Ort L wahrgenommen werden (dargestellt durch die Bedingung where L). Mittels ASK wird nach dem Service gefordert und schlussendlich durch SERV der Service angeboten, allerdings nur im Ort L. Nachdem der Client sich mit SERV synchronisiert hat, kann er sich irgendwo hinbewegen, etwas tun und letztendlich zum vorherigen Ort zurückkehren.

Wie man sieht, gehört zu jeder Aktivität SERV eine unterschiedliche Aktivität ASK und es ist leider auch nicht möglich in diesem Modell zwischen unterschiedlichen Prozessen zu unterscheiden.

In Abbildung 6 sind zwei Prozesse dargestellt, die bestimmte Services (SERV) anbieten. In der Variante links kann der Service unbeschränkt angeboten werden, in der Variante rechts allerdings nur drei mal.

MSPN vs. gewöhnliche Petri-Netze Ein statisches MSPN kann man in ein gewöhnliches Petri-Netz umformen, indem man die Örtlichkeiten, die dazu-

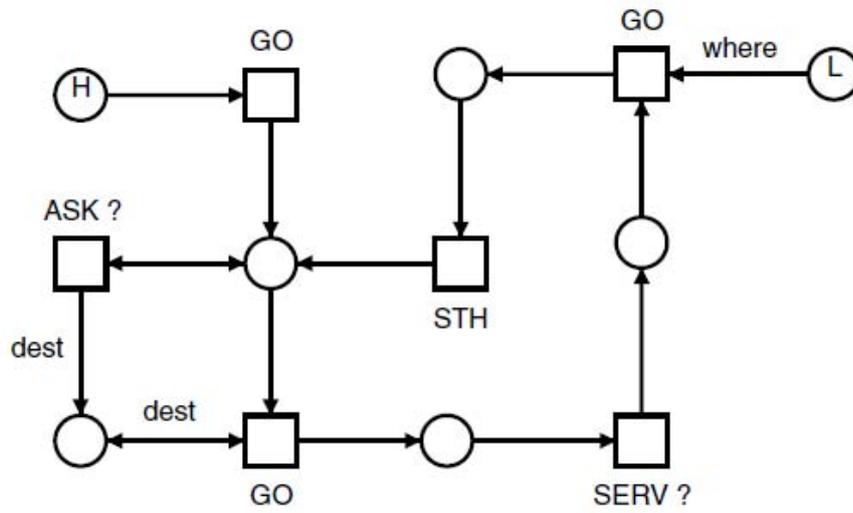


Abbildung 4. Beispiel für ein MSPN

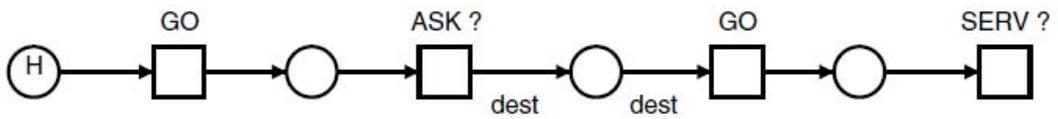


Abbildung 5.

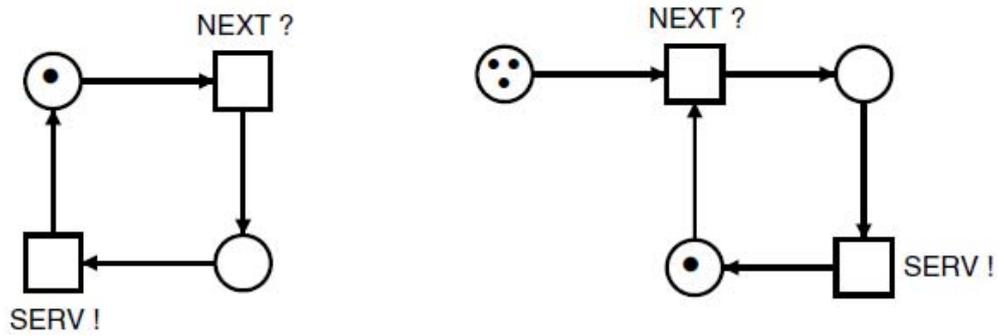


Abbildung 6. Services

gehörigen Bewegungen und die Synchronisation von Übergängen simuliert, denn es gibt in Petri-Netzen keine Orts-Token.

Mobil-synchronisierende, authentifizierende Petri-Netze: MSAPN Bisher war es nicht möglich mit MSPN die Anzahl, in der ein Prozess einen bestimmten Service in Anspruch nimmt oder den Ablauf, in dem Prozesse verschiedene Services erhalten, zu kontrollieren. Deswegen brauchen wir eine Erweiterung der MSPN.

Zunächst führen wir einen lokalen Mechanismus zur Identifikation ein. \mathbb{N} sei dabei die der Bereich der Identifikationen und da wir nicht wollen, dass Namen, die von einem Prozess erschaffen wurden, durch andere Prozesse verfälscht werden, berücksichtigen wir für jeden Prozess disjunkte Bereiche $Id = \bigoplus_{i \in I} \mathbb{N}$. Daher haben Identifizierer die Form (i, m) mit $i \in I$ (I sei die Menge der Indizes) und $m \in \mathbb{N}$. Dabei gehört jeder Index $i \in I$ zu einem einzelnen Prozess und jeder Prozess hat für jeden besitzenden Index Konstruktoren, die spezielle mit *succ* gekennzeichnete Übergänge sind, die jeweils einen Token (i, m) von einem ausgezeichneten Platz der Vorbedingung nehmen und so viele Token $(i, m + 1)$ wie die Anzahl der Nachbedingungen abgeben.

Ein MSAPN ist damit ziemlich ähnlich zu einem MSPN. Neu sind Id-Token und Id-Variablen mit zwei speziellen Variablen τ^+ und τ^- . Außerdem besitzt jedes Netz höchstens einen Übergang *succ*, welcher als Vor- und Nachbedingung einen identifizierenden Platz *Id* hat, der zuständig für das Zählen (Counter) ist, unterstützt durch die oben erwähnten zwei speziellen Variablen. Counter erkennt man daran, dass sie Vorbedingung und Nachbedingung vom Übergang *succ* sind und von keinem anderen Übergang sonst.

Modi von *succ*-Übergängen können nur (i, m) -Token verbrauchen (Id-Token) und der Wert von τ^+ muss der Nachfolger des Wertes von τ^- sein. Der Modus kann nun auch Id-Variablen auf die zugehörigen Token abbilden. Für die Spezialvariablen gilt: τ^- wird auf das Token (i, m) und τ^+ auf $(i, m + 1)$ abgebildet.

Zwei Übergänge sind genau dann kompatibel, wenn die Kompatibilitätsbedingungen der MSPN gelten und zusätzlich: Wenn eine ID-Variable (ohne Spezialvariablen!) Vorbedingung eines synchronen Übergangs ist, dann ist sie auch Vorbedingung des dazugehörigen kompatiblen Übergangs.

In Abbildung 7 haben wir nun das vorherige Beispiel aus dem MSPN-Kapitel zu einem MSAPN erweitert. Nun gibt der Absender nicht nur den Ort preis, an dem der Prozess den Service in Anspruch nehmen kann, sondern auch einen einzigartigen Identifizierer, so dass der einzige fähige Prozess, der den angebotenen Service in Anspruch nehmen kann, derjenige ist, der den selben Identifizierer vorweist. Daher muss der Prozess, der den Service anbietet, überprüfen, ob der Prozess, der beabsichtigt sich mit ihm zu synchronisieren, die entsprechende Kennzeichnung besitzt (siehe Abbildung 8).

Da Identifizierer nicht verfälscht werden können, gehört jeder einzelne von ihnen zu einem anbietenden Service. Daher macht es keinen Sinn zu versuchen, den Service mehr als einmal zu nutzen. Der Server bietet den Service an, nachdem die Identität eines Prozesses abgefragt wurde. Aber es wird nicht garantiert, dass

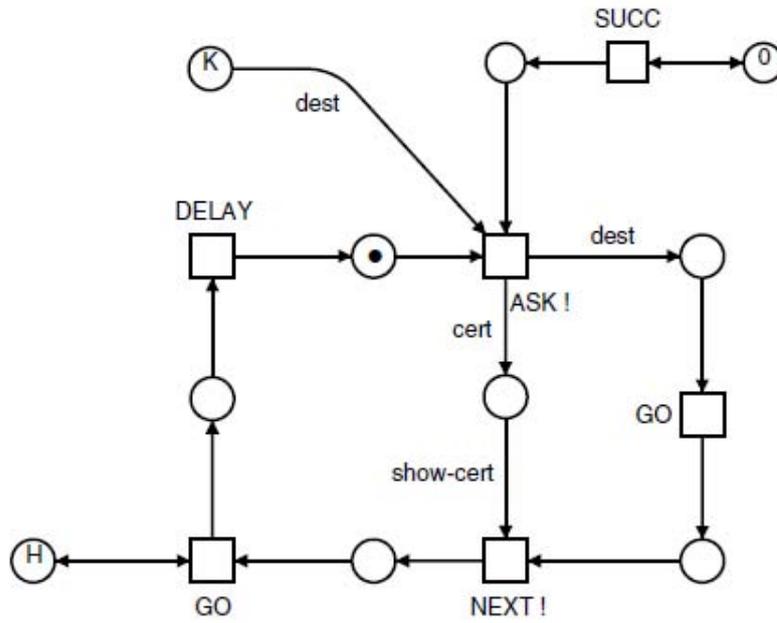


Abbildung 7. Beispiel für ein MSAPN

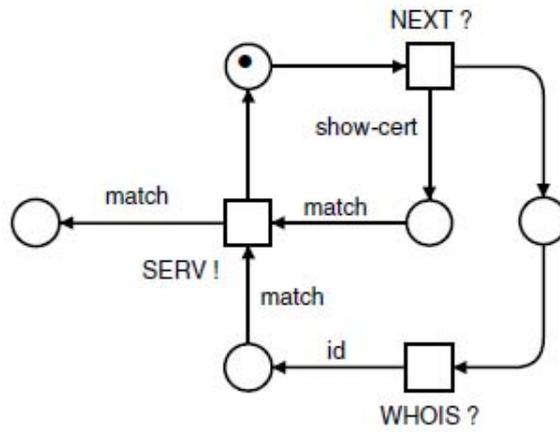


Abbildung 8. Authentifizierungsablauf

der Prozess, der den Service in Anspruch nimmt, auch derjenige ist, der seine Identität preisgegeben hat. Um das zu verhindern, gibt es die Authentifizierungs-Variablen *match*, die sicherstellen, dass der Service nur von einem Client genutzt werden kann, der die Erlaubnis hat.

4 Fazit

Das erste vorgestellte Modell, die Active Spaces, sind nützlich als Design-Tool für neue ubiquitäre Informationssysteme oder auch zum Verifizieren von gewissen Eigenschaften existierender Systeme. Ebenfalls damit möglich ist der Vergleich von Eigenschaften verschiedener Systeme. Für komplexere Systeme wird allerdings ein Modelchecker benötigt, was nichts anderes als ein Algorithmus ist, der die Wahrheit von $P \models a$ für einen gegebenen Prozess P und eine Formel a als Eingabe überprüft.

Bei Active Spaces steckt Middleware [MW] dahinter. Die Ausgangssituation ist durch die boxen-artige Darstellung besser erkennbar. Die grafische Darstellung erlaubt uns also eine Außenansicht, was den Zustand besser abschätzbar macht. Allerdings sind Prozesse in Ambients nicht vorhersehbar und man hat keinen Zeitfaktor (wenn man z. B. einen Alarm auslösen will, wenn das System seit 10 Sekunden im Fehlerzustand ist, ist dies nicht möglich, da man den Faktor Zeit nicht in diesem Modell integriert hat).

Da ubiquitäre Informationssysteme nichts anderes sind als eine Menge von Komponenten-Netzen, die Services anbieten und fordern, eignet sich das zweite Modell, die MSPN bzw. deren Erweiterung, besser. Dort wird das Anbieten und Fordern durch die synchronen Übergänge modelliert. Die Mobilität wird mittels der Bewegungsübergänge modelliert.

Und durch die Erweiterung der Sicherheit mittels eines lokalen Identifikationsmechanismus ist dieses Modell, vor allem wegen der Einfachheit ein guter Rahmen zur Spezifikation mobiler Systeme. Zwar würden gewöhnliche Petri-Netze auch ausreichen, nur diese sind zu komplex in ihrer Struktur und Größe.

Da die MSPN sich näher an den Automaten orientieren, eignet sich dieses Modell besser zur Ausführung von Prozessen. Man sieht was im jetzigen Zeitpunkt passiert und ebenfalls was als nächstes passieren wird. Die grafische Darstellung zeigt uns die genauen Abläufe im System. Aus diesem Grund ist dieses Modell für das klassische Engineering wie zugeschnitten. Es ist perfekt für die Einführung bzw. die Erzeugung von Systemen.

In beiden Modellen wird zwischen den verschiedenen Anwendern unterschieden, die am System teilnehmen. Die Örtlichkeiten spielen ebenfalls eine wichtige Rolle. Diese zwei Dinge zu modellieren, ist die Basis für ubiquitäre Informationssysteme.

Literatur

- [Mob] *Communication Primitives for Ubiquitous Systems or RPC Considered Harmful*, Umar Saif, David J. Greaves

- [UbiCWei] *The Computer for the 21st Century*, Mark Weiser,
<http://www.cc.gatech.edu/fac/Gregory.Abowd/hci-resources/area-bok/papers/p94-weiser.pdf>, 18.01.2009
- [UbiCompEn] *Ubiquitous Computing*, Volkmar Pipek, 17.12.2008,
<http://www.oldenbourg.de:8080/wi-enzyklopaedie/lexikon/technologien-methoden/Rechnernetz/Ubiquitous-Computing>
- [UbiCompTeco] *Ubiquitous Computing*, Christian Decker, 17.12.2008
<http://www.teco.edu/lehre/ubiqws0809/pdf/01introf.pdf>
- [UbiCompWiki] *Ubiquitous Computing*, 17.12.2008,
http://de.wikipedia.org/wiki/Ubiquitous_Computing
- [AdHocWiki] *Ad-hoc-Netz*, 17.12.2008,
http://de.wikipedia.org/wiki/Mobiles_Ad-hoc-Netz
- [AdHocLex] *Ad-hoc-Netzwerk*, 17.12.2008,
<http://www.itwissen.info/definition/lexikon/Ad-hoc-Netzwerk-ad-hoc-network.html>
- [VerSysWiki] *Vermaschtes Netz*, 17.12.2008,
http://de.wikipedia.org/wiki/Vermaschtes_Netz
- [AcSpPap] *Provably Correct Pervasive Computing Environments*, Anand Ranganathan, Roy H. Campbell
- [UCKon] *UbiComp Konzept: Kontext*,
<http://www.teco.uni-karlsruhe.de/lehre/ubiqws0203/10kontext.pdf>, 18.01.2009
- [PetriWeb] *Petri-Netze*, Kay Marczinik, 19.12.2008,
<http://kik.informatik.fh-dortmund.de/petrinetze/>
- [PetriWiki] *Petri-Netz*, 19.12.2008,
<http://de.wikipedia.org/wiki/Petri-Netz>
- [MSPNPap] *Mobile Synchronizing Petri Nets: A Choreographic Approach for Coordination in Ubiquitous Systems*, Fernando Rosa-Velardo, Olga Marroquín-Alonso, David de Frutos-Escrig, 2006
- [MPN] *Mobile Petri Nets*, A. Asperti, N. Busi, 1996
- [APN] *Ambient Petri Nets*, Olga Marroquín-Alonso, David de Frutos-Escrig, 2003
- [MW] *Middleware*, 18.01.2009,
<http://de.wikipedia.org/wiki/Middleware>

Probabilistisches Model Checking*: Anwendung in Ubiquitären Computersystemen

Tobias Cichos

TecO (Telecooperation Office), University of Karlsruhe Vincenz-Priessnitz-Str. 3,
76131 Karlsruhe, Germany
tobias.cichos@stud.uni-karlsruhe.de
<http://www.teco.edu>

Zusammenfassung In dieser Seminararbeit wird das Prinzip des Probabilistischen Modell Checking erklärt und die wichtigsten Grundlagen dazu definiert: Zeitdiskrete Markow-Ketten (DTMCs), indeterministische Markow-Ketten (MDP) und die probabilistische temporale Logik PCTL (Probabilistic Computation Tree Logic). Auch die Herkunft des Modell-Begriffs wird anhand verschiedener Logiken erklärt.

Anschließend wird ein typisches UbiComp-Beispiel (“active badge“: location tracking im Universitätsgebäude) behandelt. Dabei wird erst ein einfaches Beispiel gemacht, daß genau die vorher definierten Grundlagen benutzt. Dann wird noch ein komplexerer Fall des Modell Checking beschrieben, der fortgeschrittenere Techniken benutzt: PEPA, eine Modellierungssprache für zeitkontinuierliche Markow-Ketten (CTMCs), die probabilistische Logik PML und den Modellprüfer PRISM. Die Formeln dieses Beispiels werden verständlich gemacht, ohne vorher die formalen Grundlagen behandelt zu haben.

Key words: Modell Checking, Modellprüfung, Modell-Begriff, PCTL, Probabilistische Logik, Markow-Kette, Markov-Chain, Markov-Chain, Markov-Kette, active-badge, UbiComp, Ubiquitäre Systeme

1 Einleitung

Die Verlässlichkeit von ICT-Systemen ist eine Schlüsselaufgabe bei Entwurf und Implementierung. [Katoen] Viele Beispiele von Systemdefekten zeigen, daß Qualitätsmängel zu teuren Folgekosten führen können: Ariane 5 Rakete, Toll Collect Deutschland, Flughafen Denver usw.. Da die meisten Fehler bei Entwurf und Implementierung eines Systems einfließen [4], sollte die Systemverifikation Fehler möglichst früh im Produktionsprozess erkennen. Während das Testen als Methode der Fehlersuche erst im fortgeschrittenen Stadium der Fertigstellung möglich ist, kann dazu neben einfachem wiederholtem gedanklichen Durchgehen des Konzeptes auch die Simulation dienen.

Eine Modell-basierte Verifikationstechnik ist das Modell Checking, bei dem nicht direkt das System, sondern eine abstrahierte Version davon maschinell auf die

* engl. Modellprüfung

gewünschten Eigenschaften geprüft wird.
 Für UbiComp-Systeme¹ ist die große Herausforderung die Zusammenführung von stofflicher Welt (Real World) und digitaler Informationsverarbeitung (Virtual World), siehe Abbildung 1. Daher ist es beim Modell Checking ubiquitärer

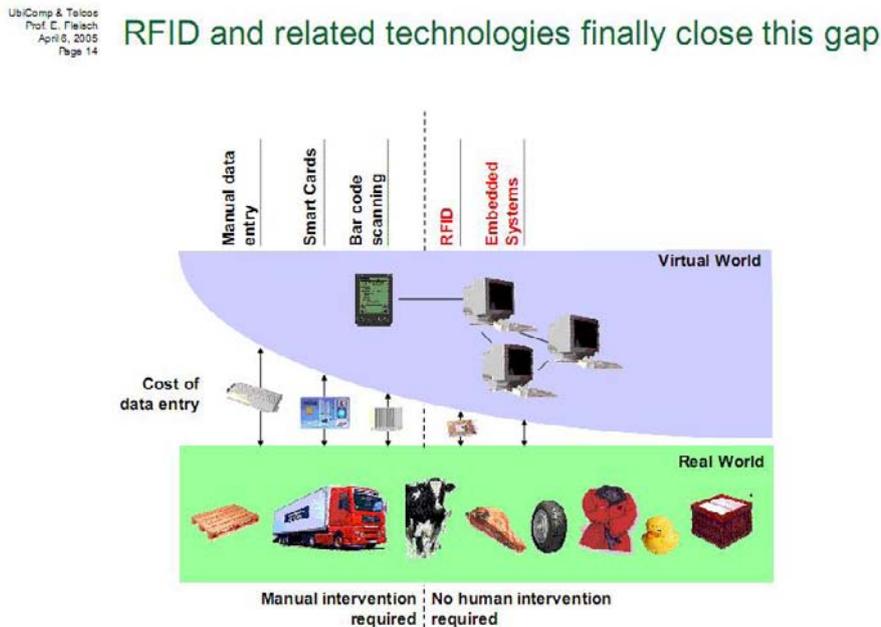


Abbildung 1. UbiComp-Technologien schlagen die Brücke zwischen stofflicher Welt und EDV

Systeme wichtig, daß alle beteiligten (auch externen) Komponenten Teil des zu prüfenden Modells sind. Wegen den den Unregelmäßigkeiten, die die verwendeten Funkverbindungen mitbringen (wegen Abschattung usw.) eignen sich Modelle Logiken, die die entsprechenden Wahrscheinlichkeiten einbringen. Es handelt sich dabei um Markow-Ketten und probabilistische Logiken wie die PCTL.

2 Probabilistisches Model Checking [Katoen]

Die Grafik 2 zeigt die einzelnen Schritte der Modellprüfung:
 In der Modellierungsphase werden einerseits das Verhalten des Systems in exakter, eindeutiger Weise beschrieben. Die Abstrahierung geschieht meist in Form eines endlichen Automaten, wobei eine gewisse Ungenauigkeit toleriert wird. Die Sprache der Formalisierung hängt vom bevorzugten Modell Checker ab, dem Computerprogramm das die eigentliche Modellprüfung durchführt (und evtl.

¹ von engl. Ubiquitous Computing (Wortschöpfung von Marc Weiser, Xerox Park): Ubiquitäre(allgegenwärtige) Computersysteme

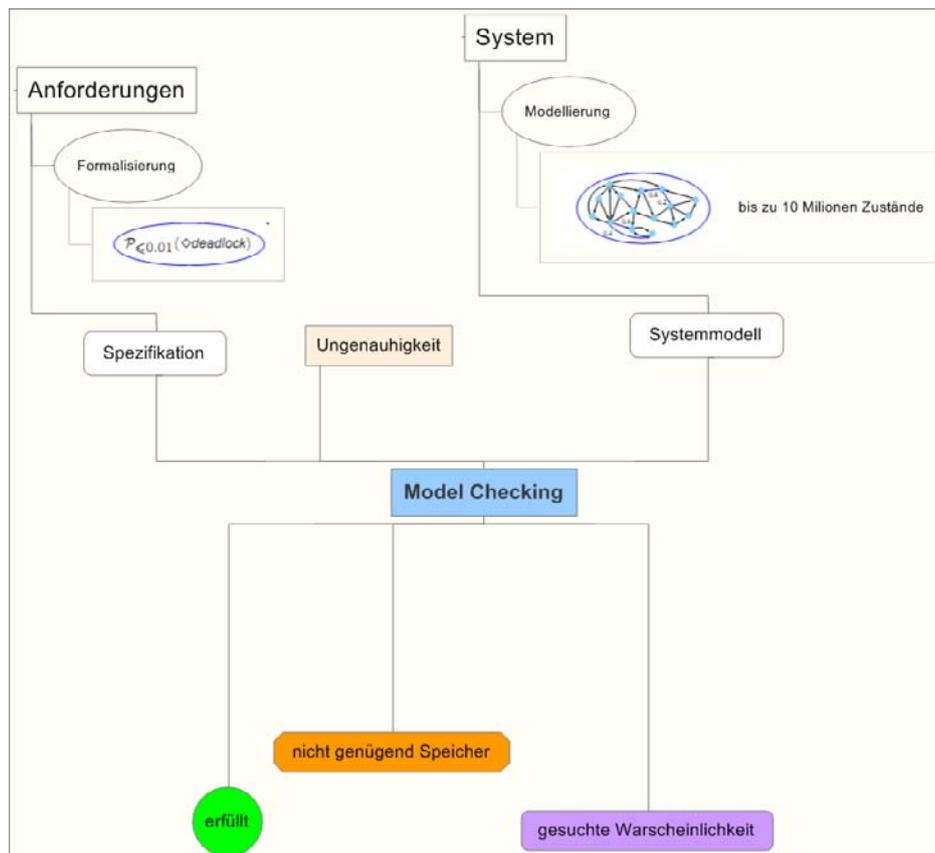


Abbildung 2. Schema der Modellprüfung (Model Checking)

eine Simulation erlaubt, die das Auffinden etwaiger Fehler unterstützt). Ein prominenter Modell Checker ist z.B. PRISM, der die Formalisierung in der PRISM-Language oder in PEPA erlaubt.

Andererseits werden die Anforderungen mit Hilfe einer passenden Logik z.B. PCTL zur Spezifikation formuliert. In der eigentlichen Modell Checking Phase wird die zu betrachtende Eigenschaft systematisch für jeden Zustand des Modells auf Gültigkeit geprüft.

Es gibt grundsätzlich drei mögliche Ergebnisse der Modellprüfung: Die formulierte Eigenschaft ist entweder zutreffend oder nicht, oder das Modell war zu umfangreich für die zu Verfügung stehende Rechenleistung. War das Ergebnis positiv, so kann die nächste Formel getestet werden und so z.B. eine Annäherung für die Wahrscheinlichkeit eines bestimmten Ereignisses wie ein Deadlock gefunden werden.

3 Probabilismus in der Modellierung Ubiquitärer Systeme

Folgende Eigenschaften Ubiquitärer Systeme sprechen bei einer sinnvollen Systemspezifikation für eine probabilistische Modellierung:

- Objekt- oder Kontexterkenkung erlaubt keine hundertprozentigen Garantien,
- Funkverbindungen werden unvorhersehbar von ihrer Umgebung beeinflusst: Abschattung, Beugung, Reflexion, Abschirmung.
- Es werden zum Teil probabilistische Kollisionsauflösungsverfahren verwendet: TDMA, Aloah
- Die Wahrscheinlichkeitstheorie kann Indeterminismen umgehen (probabilistische Aussagen vs. Indeterminismen)
Indeterminismen bestehen bei:
 - dem Zugriff von mehreren Geräten auf einen Flaschenhals eines Servers oder Sensors
- Ubiquitäre Systeme stellen oft einen Mehrwert einer Appliance² dar, der auch ausnahmsweise einmal ausfallen darf, ohne daß eine Katastrophe ausbricht. Die Frage ist dabei, wie teuer ist ein "Ausfall", d.h. wie schlimm ist es, wenn ein Ergebnis unterbleibt?

4 Der Modell-Begriff

4.1 In der Aussagenlogik

In der Aussagenlogik betrifft der Modell-Begriff allein die Interpretation über einer aussagenlogischen Signatur.

² engl. Haushaltsgerät bzw. Gebrauchsgegenstand

Es sei Σ eine aussagenlogische Signatur (Menge von Symbolen für die Elementaraussagen).

Eine Interpretation I über Σ ist eine beliebige Abbildung

$$I : \Sigma \longrightarrow \{W, F\}.$$

Eine Interpretation der Aussagenlogik ordnet also jeder Elementaraussage einer Formel einen Wahrheitswert zu. Die weitere Zuordnung von Wahrheitswerten zu allen aus Elementaraussagen und Operatoren zusammengesetzten Formeln $val_I : For_\Sigma \longrightarrow \{W, F\}$ wird Auswertung über Σ genannt. Es gilt

$$\begin{aligned} val_I(1) &= W \\ val_I(0) &= F \\ val_I(P) &= I(P) \text{ für jedes } P \in \Sigma \\ val_I(\neg A) &= \begin{cases} F & \text{falls } val_I(A) = W, \\ W & \text{falls } val_I(A) = F \end{cases} \\ val_I(A \wedge B) &= \begin{cases} W & \text{falls } A=W \text{ und zugleich } B=W, \\ F & \text{sonst} \end{cases} \\ & \text{usw.} \end{aligned}$$

Definition: Ein *Modell* einer aussagenlogischen Formel F ist eine Interpretation, so daß $val_I(F) = W$.

(*Intuitiv:* Ein *Modell* einer Formel ist eine Belegung der vorkommenden Elementaraussagen mit einem Wahrheitswert, so daß die Formel bei ihrer Auswertung den Wahrheitswert wahr erhält.)

4.2 In der Modallogik

Im Vergleich zur einfachen Aussagenlogik, wo ein Modell nur eine Belegung der Elementaraussagen ist, gehören in der Modallogik zum Modell der gesamte Zustandsraum, die Zustandsübergangsbeschreibung und die Wahrheitsinterpretation der modallogischen Formeln für jeden einzelnen Zustand.

Eine solche (Kripke-)Struktur ist dann ein Modell einer Formel, wenn diese Formel in jedem Zustand der Struktur wahr ist.

4.3 In der Temporallogik (LTL, CTL, PCTL)

Ein Modell einer temporallogischen Formel ist eine ebenso umfassende Struktur wie in der Modallogik. Der Unterschied besteht darin, daß die Formel nicht grundsätzlich für alle Zustände der Struktur wahr sein muß, sondern die Formeln beinhalten eine genaue Angabe, für welche Zustände sie oder eine Teilformel von ihr gelten soll.

5 Markow-Ketten

Markow-Ketten verhalten sich wie gewöhnliche Zustandsübergangsdiagramme, mit der Ausnahme, daß an Stelle von indeterministischen Verzweigungen diese mit Wahrscheinlichkeiten versehen werden. Der Folgezustand eines Zustandes, sagen wir s , wird anhand einer vorgegebenen Wahrscheinlichkeitsverteilung bestimmt, die nur von s abhängt (und nicht etwa von dem Pfadstück, das von dem Startzustand zu dem Zustand s geführt hat). Dementsprechend ist jede Weiterentwicklung des Systems nicht abhängig vom bisherigen Verlauf, sondern nur vom aktuellen Zustand. Dieser Sachverhalt nennt sich auch die “memoryless property“ (Gedächtnislosigkeit).

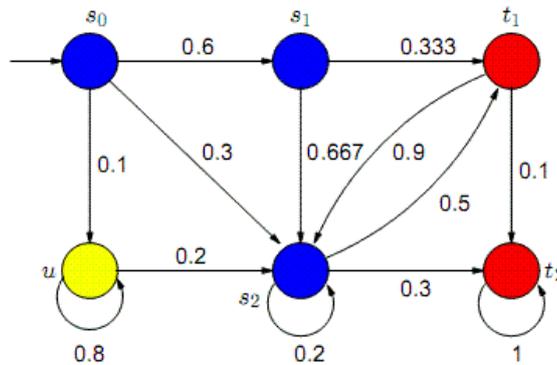


Abbildung 3. Beispiel einer Markow-Kette

5.1 Definition (zeitdiskrete) Markow-Kette (MK)

Eine (zeitdiskrete) Markow-Kette ist ein Tupel $\mathcal{M} = (S, P, \nu_{init}, AP, L)$ mit

- S ist eine abzählbare, nicht leere Zustandsmenge,
- $P : S \times S \rightarrow [0, 1]$ ist die *Übergangswahrscheinlichkeits-Funktion*, so daß für jeden Zustand s gilt:

$$\sum_{s' \in S} P(s, s') = 1,$$

- $\nu_{init} : S \rightarrow [0, 1]$ ist die Initialverteilung, so daß

$$\sum_{s \in S} \nu_{init}(s) = 1, \text{ und}$$

- AP ist eine Menge atomarer Aussagen und $L : S \rightarrow 2^{AP}$ eine Markierungsfunktion.

\mathcal{M} wird *endlich* genannt, wenn S und AP endlich sind.

Für endliche \mathcal{M} ist die *Größe* von \mathcal{M} die Anzahl der Zustände addiert zur Anzahl der Paare $(s, s') \in S \times S$ mit $P(s, s') > 0$.

Hinweis: Eine Markow-Kette ist (im allgemeinen) gar keine Kette im herkömmlichen Sinn. Sondern sie kann Verzweigungen und auch Zyklen enthalten.

Die Zustands-Übergangswahrscheinlichkeits-Funktion bestimmt für jedes Paar (s, s') von Zuständen die Wahrscheinlichkeit $P(s, s')$, daß auf s in einem Schritt s' folgt, d.h. durch einen einzigen Zustandsübergang.

Der Wert $\nu_{init}(s)$ gibt die Wahrscheinlichkeit an, mit der das System mit dem Zustand s startet. Die Zustände s mit $\nu_{init}(s) > 0$ werden als Startzustände angesehen.

Gleichfalls sind die Zustände s' , für die $P(s, s') > 0$ ist, die möglichen Nachfolger von s .

Für den Zustand s und $T \subseteq S$ schreibt man

$P(s, T)$ für die Wahrscheinlichkeit eines Zustandsübergangs (in einem einzigen Schritt) von s nach irgendeinen Zustand aus T . Das bedeutet:

$$P(s, T) = \sum_{t \in T} P(s, t).$$

Die Zustandsübergangswahrscheinlichkeitsfunktion lässt sich identifizieren mit der Matrix

$$(P(s, t))_{s, t \in S}.$$

Dabei enthält die

Zeile $P(s, *)$ die Wahrscheinlichkeiten für die Übergänge von Zustand s zu allen Zuständen der Zustandsmenge S , während die Spalte $P(*, s)$ für alle Zustände die Wahrscheinlichkeiten angibt, daß auf sie der Zustand s folgt.

Ähnlich wird die Initialverteilung ν_{init} oft als der Vektor $(\nu_{init}(s))_{s \in S}$ betrachtet.

6 Markow Decision Prozesse

Indeterminismus ist bei Markow-Ketten nicht möglich, da dort alle Verzweigungen mit einer festgelegten Wahrscheinlichkeit stattfinden. Markow Decision Prozesse (kurz MDP) können als Variante von Markow-Ketten angesehen werden, bei der sowohl probabilistische als auch indeterministische Entscheidungen möglich sind.

Wie bei Markow-Ketten, können die probabilistischen Entscheidungen dienen zur Modellierung und Quantifizierung von zufallsbestimmten Ereignissen wie

beim Werfen eines Würfels oder das Senden einer Nachricht über einen Kanal, bei dem Datenverlust auftreten kann. Auch für die Abbildung des Interface eines Systems mit seiner Umgebung kann die Verwendung von Wahrscheinlichkeiten sinnvoll sein. Bei einem Snack-Automat zum Beispiel könnte es passend sein, die Wahrscheinlichkeit des Verkaufs eines Schoko-Riegels mit $\frac{8}{10}$ anzugeben und die eines Apfels mit $\frac{2}{10}$. Dies macht jedoch statistische Untersuchungen notwendig die für das Verhalten der Umgebung (z.B. der Snack-Käufer) Durchschnittswerte für die Wahrscheinlichkeitsverteilung liefern. Sind diese Informationen nicht verfügbar, oder müssen Zusicherungen gemacht werden, die für alle möglichen Umgebungen gelten, so liegt es nahe, auf indeterministische Verzweigungen zurückzugreifen.

Ansonsten kommt bei entsprechenden zufallsbestimmten Algorithmen und bei der Abstraktion von Markow-Ketten, d.h. der Zusammenfassung einzelner Zustände, Indeterminismus zum Einsatz.

6.1 Definition Markow Decision Prozess (MDP)

Eine Markow Decision Prozess ist ein Tupel $\mathcal{M} = (S, Act, P, \nu_{init}, AP, L)$ mit

- S ist eine abzählbare Zustandsmenge,
- Act ist eine Mengen von Aktionen,
- $P : S \times Act \times S \rightarrow [0, 1]$ ist die *Übergangswahrscheinlichkeits-Funktion*, so daß für jeden Zustand s und Aktion $\alpha \in Act$ gilt:

$$\sum_{s' \in S} P(s, \alpha, s') \in \{0, 1\},$$

- $\nu_{init} : S \rightarrow [0, 1]$ ist die Initialverteilung, so daß

$$\sum_{s \in S} \nu_{init}(s) = 1, \text{ und}$$

- AP ist eine Menge atomarer Aussagen und $L : S \rightarrow 2^{AP}$ eine Markierungsfunktion.³

Eine Aktion ist *aktiviert im Zustand* s genau dann, wenn $\sum_{s' \in S} P(s, \alpha, s') = 1$. $Act(s)$ bezeichne die Menge der in s aktivierten Aktionen.

Für jeden Zustand s muß gelten: $Act(s) \neq \emptyset$.

Jeder Zustand s' , für den gilt: $\sum_{s' \in S} P(s, \alpha, s') < 1$ wird α -Nachfolger von s genannt.

³ Als Markierung wird gewöhnlich eine atomare Aussage benutzt, die für diesen Zustand bezeichnend ist oder es wird umgekehrt die Markierung hergenommen als die für den Zustand geltende atomare Aussage. Dabei werden Markierung und Aussage identifiziert: Aus der für jeden Zustand vorhandenen atomaren Aussage "Die Markierung des Zustands ist x " wird " x ", wobei die Markierung $x := L(s)$ für jeden Zustand s durch die Markierungsfunktion $L(s)$ gegeben ist.

7 PCTL

Die Probabilistic Computation Tree Logic⁴ (kurz PCTL) ist eine temporale Logik, basierend auf der Logik CTL (Computation Tree Logic). Eine PCTL-Formel formuliert Bedingungen an einen Zustand einer Markow-Kette. Die Interpretation ist wahr oder falsch. Die PCTL erweitert die CTL um Möglichkeit, für eine Aussage über einen Pfad zusätzlich Bedingungen an die Wahrscheinlichkeit anzugeben, mit der ein Pfad diese Aussage erfüllt.

7.1 Syntax der PCTL

Die Syntax der PCTL ist zweischichtig aufgebaut: aus Zustandsformeln und Pfadformeln. Zustandsformeln machen eine Aussage über Zustände und Pfadformeln gelten für Pfade.

$J \subseteq [0,1]$ sei ein Intervall mit rationalen Schranken.

$\mathbb{P}_J(\varphi)$ gilt für einen Zustand, wenn die Wahrscheinlichkeit, daß ein von diesem Zustand ausgehender Pfad φ erfüllt, in dem Intervall J ⁵ liegt.
6

AP sei die Menge der atomaren Aussagen und
 $a \in AP$ eine atomare Aussage;

Φ ist eine PCTL-Zustandsformel über AP . Dabei gilt folgende Grammatik:

$$\Phi ::= true \mid a \mid \Phi_1 \wedge \Phi_2 \mid \neg\Phi \mid \mathbb{P}_J(\varphi)$$

Dabei sind
 Φ_1 und Φ_2 ebenfalls Zustandsformeln und
 φ eine Pfadformel: Sei

⁴ wörtlich übersetzt Wahrscheinlichkeitstheoretische Berechnungs-Baum-Logik

¹ Anstatt das Intervall J explizit in eckigen Klammern anzugeben, wird meist eher eine Abkürzung verwendet; z.B. steht $\mathbb{P}_{\leq 0,5}(\varphi)$ für $\mathbb{P}_{[0;0,5]}(\varphi)$, $\mathbb{P}_{=1}(\varphi)$ für $\mathbb{P}_{[1;1]}(\varphi)$ und $\mathbb{P}_{\gg 0}(\varphi)$ für $\mathbb{P}_{]0;1]}(\varphi)$.

⁶ Bezieht sich die Formel auf eine Menge von Zuständen, so handelt es sich entsprechend um alle Pfade, die von einem Zustand der Menge ausgehen.

Ohne Angabe, für welchen Zustand die Formel gelten soll, bezieht sie sich auf den Startzustand, bzw. die Menge der Initialzustände. Es handelt sich dann also um **alle** Pfade des Graphen.

Im Vergleich mit der CTL ersetzt dieser Wahrscheinlichkeits-Operator das einfachere Konzept mit \forall (für alle Pfade) und \exists (für mindestens einen Pfad):

In einem Zustand gilt die Formel

$\exists\varphi$, wenn es irgendeinen Pfad gibt, der mit diesem Zustand beginnt und für den die Formel φ gilt. Entsprechend gilt in einem Zustand die Formel
 $\forall\varphi$, wenn die Formel φ für alle Pfade gilt, die in diesem Zustand beginnen.

$n \in \mathbb{N}$ eine natürliche Zahl.

Eine PCTL-Pfadformel über AP ist aufgebaut nach folgender Grammatik:

$$\varphi ::= \bigcirc\Phi \mid \Phi_1 \cup \Phi_2 \mid \Phi_1 \cup^{\leq n} \Phi_2$$

$\bigcirc\Phi$ gilt für einen Pfad in einem Zustand, wenn Φ im nächsten Zustand im Pfad gilt. “ \bigcirc “ wird “*next-step-Operator*“⁷ genannt.

$\Phi_1 \cup \Phi_2$ gilt für einen Pfad, wenn dieser einen Zustand enthält, in dem Φ_2 gilt, und zugleich Φ_1 für dessen sämtliche Vorgängerzustände gilt. “ \cup “ ist der “*until-Operator*“⁸.

$\Phi_1 \cup^{\leq n} \Phi_2$ bedeutet, daß ein Φ_2 -Zustand innerhalb n Zustandsübergängen erreicht wird, und daß alle Zustände vor Erreichen des Φ_2 -Zustandes Φ_1 erfüllen. $\Phi_1 \cup^{\leq n} \Phi_2$ ist die Schrittzahl-begrenzte Variante von $\Phi_1 \cup \Phi_2$.

Daraus folgt, daß die Operatoren \bigcirc und \cup (und seine gebundenen Variante) immer im Argument von \mathbb{P} stehen.

Andere aussagenlogische Operatoren werden in der üblichen Weise abgeleitet.

Beispielsweise erhält man

$$\Phi_1 \vee \Phi_2 \text{ aus } \neg(\neg\Phi_1 \wedge \neg\Phi_2).$$

Der *Diamond*⁹-Operator \diamond leitet sich so ab:

$$\diamond\Phi = true \cup \Phi$$

$\diamond\Phi$ gilt für einen Pfad, wenn er einen Zustand enthält, für den Φ gilt.

Ebenso schreiben wir für die schrittzahl-begrenzte Variante $\diamond^{\leq n}$:

$$\diamond^{\leq n}\Phi = true \cup^{\leq n} \Phi$$

Ein Pfad erfüllt also $\diamond^{\leq n}\Phi$, wenn Φ nach dem spätestens n -ten Schritt erfüllt wird.

Für den *always*¹⁰-Operator \square gelten folgende Ableitungen:

$$\square\Phi = \neg(\diamond\neg\Phi)$$

$$\mathbb{P}_{\leq p}(\square\Phi) = \mathbb{P}_{\geq 1-p}(\diamond\neg\Phi)$$

$$\mathbb{P}_{\geq p}(\square\Phi) = \mathbb{P}_{\leq 1-p}(\diamond\neg\Phi)$$

³ von engl. “nächster Schritt“

⁸ von engl. “bis“

⁹ von engl. “Diamant“

¹⁰ von engl. “immer“, im Sinne von “für alle Zustände“

$\Box\Phi$ gilt für einen Pfad, wenn für alle seine Zustände Φ gilt.

Für die schrittzahl-begrenzte Variante gilt der Zusammenhang:

$$\begin{aligned}\Box^{\leq n}\Phi &= \neg(\Diamond^{\leq n}\neg\Phi) \\ \mathbb{P}_{[p;q]}(\Box^{\leq n}\Phi) &= \mathbb{P}_{[1-q;1-p]}(\Diamond^{\leq n}\neg\Phi)\end{aligned}$$

Ein Pfad erfüllt also $\Box^{\leq n}\Phi$, wenn Φ wenigstens für die ersten n Schritte erfüllt bleibt.

7.2 Beispiele

Die Formel

$$\bigwedge_{1 \leq i \leq 6} \mathbb{P}_{=\frac{1}{6}}(\Diamond i)$$

drückt für das Modell eines 6-seitigen Spielwürfel aus, daß alle Ausgänge gleich wahrscheinlich sein sollen. Da es 6 Endzustände gibt, muß für diese jeweils die Wahrscheinlichkeit $\frac{1}{6}$ gelten.

Stellen wir uns nun ein Kommunikationsprotokoll vor, das einen fehlerhaften Kanal benutzt, so daß Paketverlust auftreten kann: Die PCTL-Formel

$$\mathbb{P}_{=1}(\Diamond \text{Auslieferung}) \wedge \mathbb{P}_{=1}\left(\Box(\text{Sendeversuch} \longrightarrow \mathbb{P}_{\geq 0,99}(\Diamond^{\leq 3} \text{Auslieferung}))\right)$$

sichert zu, daß fast mit Sicherheit irgendeine Nachricht ausgeliefert werden wird (erste Klausel) und, daß fast mit Sicherheit auf jeden Versuch eine Nachricht zu senden innerhalb von 3 Schritten mit einer Wahrscheinlichkeit größer oder gleich 0,99 eine Auslieferung erfolgt.

8 Fallstudie: Ein Ortungs-System

Als Beispiel betrachten wir hier die Aufgabe, ein Ortungs-System zu modellieren, bei dem der Aufenthaltsort von Personen und Ausrüstung innerhalb eines Gebäudes von einem zentralen Kontrollsystem mitverfolgt (überwacht) wird. Ein solches System wird in Erwägung gezogen für das James Clerk Maxwell Gebäude an der Universität von Edinburgh. Das Gebäude ist dafür berüchtigt, daß es sehr verwirrend ist, seinen Weg zu darin finden. Daher wäre das Ortungs-System sehr nützlich dabei, Besucher, die in dem Labyrinth von Gängen verloren gehen, wiederzufinden. Außerdem würde das System Sekretärinnen und Sekretären dabei helfen, ihre Professoren wiederzufinden, die in jeglichem Lehr- und Konferenz-Saal oder in einem Büro von Kollegen sein können. Gleichzeitig wäre es z.B. für die Jagd nach den Leih-Notebooks ohne Netzwerkverbindung, die für die ausspäh-sichere Vorbereitung von Klausuren ausgeliehen werden können, eine unschätzbare Hilfe.

Derartige Lokationssysteme werden umgesetzt mit Hilfe von sogenannten active badges¹¹. Das sind Geräte der Größe einer Scheckkarte, die einzelne Infrarot-Signale übermitteln, die von vernetzten Sensoren erkannt werden. Solche Systeme werden bereits genutzt von verschiedenen europäischen Universitäten und von Forschungslaboren in den USA.

Die Universität von Karlsruhe gibt zu Anfang jeden Studienjahrs "intelligente" Immatrikulations-Karten heraus, die sowohl als elektronisches Zahlungsmittel als auch als programmierbarer Türschlüssel dienen. In Edinburgh sollen Nachfolger solcher Karten auch dem Location-Tracking¹² dienen. Die Lebensdauer der Batterie eines solchen Gerätes hat sich typischerweise als ein Jahr erwiesen [7]. Es ist daher notwendig, die Leistung des Gerätes so einzustellen, daß die Häufigkeit, mit der die Registrierung ausgeführt wird, gering genug ist um Batterie zu sparen, während andererseits oft genug gesendet werden muß, so daß in der Datenbank des Systems immer der genaue Aufenthaltsort gespeichert ist. [2]

Ein einfachstes Modell für dieses Szenario ist eine Markov-Kette, die den Weg der ID-Übertragung von der Karte bis in den Datenspeicher darstellt. Bei diesem Modell soll zunächst nur die Aktivität für einen einzelnen Menschen, der die Karte mit sich führt, bei einem Sendeversuch betrachtet werden: Das Senden der Karte wird als sicher angenommen, so daß der erste Zustand unserer Markov-Kette mit "Karte sendet" benannt ist.

Für die Initialverteilung gilt also:

$$\iota_{init}(\text{"Karte sendet"}) = 1.$$

Bei der Funkübertragung kann durch die unterschiedlichen Situationen, in der die Karte ist, die Übertragung beeinflusst werden. So daß der Empfang der Nachricht am Sensor nicht zu 100 % zugesichert werden kann. Den Folgezuständen "Sensor empfängt" und "Sensor empfängt nicht" ordnet daher die Übergangswahrscheinlichkeits-Funktion P verschiedene Wahrscheinlichkeiten zu:

$$\begin{aligned} P(\text{"Karte sendet"}, \text{"Sensor empfängt"}) &= p_1, \\ P(\text{"Karte sendet"}, \text{"Sensor empfängt nicht"}) &= 1 - p_1 \end{aligned}$$

Als nächstes soll die Sensorlogik die (normalerweise)empfangene ID über das LAN weiter senden zum Datenbankserver. Hierbei können wiederum, wenn auch nur in seltenen Fällen, folgende Probleme auftreten:

- Im Netzwerk tritt in einem Überlastfall Paketverlust auf, der aufgrund der Realtime-Eigenschaften der Sensorlogik nicht ausgeglichen wird, da dann sowieso schon die nächste ID-Sendung an der Reihe ist.
- Der Server könnte überlastet sein. Für beide Fälle ist wieder eine gemeinsame Fehlerwahrscheinlichkeit $1-q$ festzulegen. Wobei q die Wahrscheinlichkeit für ein einwandfreies Empfangen ist. Formalisiert für unsere Markov-Kette schreibt sich

¹¹ engl. aktive Abzeichen

¹² engl. Verfolgung des Aufenthaltsortes

dies:

$$\begin{aligned} P(\text{“Sensor empfängt“}, \text{“Speicherung in DB“}) &= p_1, \\ P(\text{“Sensor empfängt“}, \text{“keine Speicherung in DB“}) &= 1 - p_1 \\ P(s, s') &= 0 \text{ für alle übrigen Zustands-Paarungen}(s, s') \end{aligned}$$

Unsere Zustandsmenge S ist in diesem Fall: $S = \{\text{Karte sendet, Sensor empfängt, Sensor empfängt nicht, Speicherung in DB, keine Speicherung in DB}\}$ Während die atomaren Aussagen mit den Namen der Zustände identifiziert wurden. (Beispiel: Im Zustand “Sensor empfängt“ gilt die atomare Aussage, daß der Sensor empfängt. Diese gilt nur für diesem Zustand und in diesem Zustand gilt nur diese eine Aussage.) Der Zustands- bzw. Aussagen-Raum und damit das Markow-Modell sind endlich.

Die Wahrscheinlichkeit, daß in diesem Modell irgendein Fehler passiert, beträgt $p_1 - 1 + p(q_1 - 1)$ Möchte man für dieses Modell nun beispielsweise festlegen, daß diese Wahrscheinlichkeit kleiner 1% ist, so kann lässt sich dies mithilfe der PCTL-Formel

$$\mathbb{P}_{<0,01}(\diamond \text{“Sensor empfängt nicht“} \vee \diamond \text{“keine Speicherung in DB“})$$

(Zur Erinnerung: $\diamond\Phi$ gilt für einen Pfad, wenn er einen Zustand enthält, für den Φ gilt.)

Dies war jedoch nur eine sehr einfache Einführung, um für die weiter oben ausgeführten Formalismen von Markow-Ketten und PCTL die Funktionsweise zu verdeutlichen. Eine wesentliche Dimension, in der unser Modell erweitert werden sollte, ist die zeitliche Wiederholung von Sendeversuchen mit einer bestimmten Frequenz. Für den erfolgreichen Empfang davon wird ein Erwartungswert (engl. “rate“) angegeben. Dieser bezieht sich auf die exponentialverteilte Wahrscheinlichkeitsverteilung [Katoen] und gibt die durchschnittliche Anzahl von erwünschten Ergebnissen (z.B. pro Sekunde) an. Daraus folgt, daß mit einer (normalerweise sehr) geringen Wahrscheinlichkeit auch einmal 0 Erfolge pro Zeiteinheit realisiert werden können.

Für die Modellierung dieses Sachverhaltes eignen sich beispielsweise die PRISM-Modelling-Language oder ,wie folgender Ausführung, PEPA [3]:

Der eben besprochenen Erwartungswert taucht in unserem Beispiel an drei Stellen auf: Es sei m der Erwartungswert für die Bewegung des Kartenträgers in einen anderen Gang, und r der Erwartungswert für die Bewegung des Kartenträgers in einen anderen Gang.

$$\begin{aligned} m &= 0.1; \\ r &= 2.5; \\ s &= 45.0; \end{aligned}$$

Um das Beispiel etwas überschaubar zu halten, nehmen wir ein System an, das nur einen einzigen Menschen hat. Außerdem drei Gänge, die wie ein U angeordnet sind: Gang 14 und Gang 16 liegen dabei an den Enden, Gang 15 in der Mitte. D.h. es ist möglich, von Gang 15 in die beiden anderen Gänge zu gehen, aber von Gang 14 und Gang 16 kann man direkt nur in Gang 15 wechseln. Der folgende PEPA-Text einschließlich die Angabe der rates oben ist passend codiert als die Eingabedatei für den Modellprüfer PRISM. Der erste Teil beschreibt die Bewegung des einen Menschen in den drei Gängen und die parallel laufende Registrierung des Aufenthaltsortes beim Sensor:

Das Modul besteht aus den Zuständen P14, P15, und P16. Ihre Nachfolger stehen jeweils hinter dem Punkt. Durch das Pluszeichen werden die gleichzeitigen Aktivitäten, Registrierung und Bewegung, getrennt:

$$\begin{aligned} P14 &= (reg14, r).P14 + (move15, m).P15; \\ P15 &= (reg15, r).P15 + (move14, m).P14 + (move16, m).P16; \\ P16 &= (reg16, r).P16 + (move15, m).P15; \end{aligned}$$

Da die Aktivitäten im zweiten Modul schon im ersten Modul eine rate zugewiesen bekommen, steht hier als rate “infty“ (von infinity¹³). Das bedeutet, daß hier Zustandsübergänge gleichzeitig mit Modul 1 (mit rate r) stattfinden.

$$\begin{aligned} S14 &= (reg14, infty).T14; \\ S15 &= (reg15, infty).T15; \\ S16 &= (reg16, infty).T16; \end{aligned}$$

Das dritte Modul beschreibt den Report des Sensors an die Datenbank mit rate s. Die (lokale) Initiierung erfolgt hier einfach durch die Identität von T14/T15/T16 als Folgezustände von Modul 3. Die S14 bis 16 dienen wieder der “Aufrechterhaltung der Zustands(ketten)-Auswahl“. Aber nur solange bis über reg ein anderer Zustand durch das erste Modul initiiert wird; denn es ist immer genau ein Zustand aktiv.

$$\begin{aligned} T14 &= (rep14, s).S14; \\ T15 &= (rep15, s).S15; \\ T16 &= (rep16, s).S16; \end{aligned}$$

Im “Datenbankmodul“ wird dann der Zustand aktiviert, der der Gangnummer entspricht:

$$\begin{aligned} DB14 &= (rep14, infty).DB14 + (rep15, infty).DB15 + (rep16, infty).DB16; \\ DB15 &= (rep14, infty).DB14 + (rep15, infty).DB15 + (rep16, infty).DB16; \\ DB16 &= (rep14, infty).DB14 + (rep15, infty).DB15 + (rep16, infty).DB16; \end{aligned}$$

¹³ engl. unendlich

Das System wird beispielsweise initialisiert in Gang 14 (und DB14) und außerdem werden noch einmal die gemeinsamen Aktivitäten definiert durch:

$$(P14 < reg14, reg15, reg16 > (S14 <> S15 <> S16)) < rep14, rep15, rep16 > DB14$$

Aufgrund dieser Modell-Beschreibung ist es dem Modellprüfer möglich, Aussagen über das System zu machen. Diese können in einer zu CTMCs passenden Logic wie die für den PRISM Model Checker benötigte CTL (classical temporal logic) oder der an PEPA angelehnten wahrscheinlichkeitstheoretischen modalen Logic PML (Probabilistic Modal Logic) formuliert werden.

Um ein Maß für die Leistungsfähigkeit des Gesamtsystems anzugeben, formulieren wir die Anforderung, daß die Fehlerhäufigkeit für die Übereinstimmung von Datenbankeintrag und tatsächlichem Aufenthaltsort 1% nicht übersteigen darf (in PML):

$$\begin{aligned} & Pr(reg_{14} \uparrow \wedge \neg(DB \equiv DB_{14})) \\ + & Pr(reg_{15} \uparrow \wedge \neg(DB \equiv DB_{15})) \\ + & Pr(reg_{16} \uparrow \wedge \neg(DB \equiv DB_{16})) \geq 0,01 \end{aligned}$$

9 Ausblick

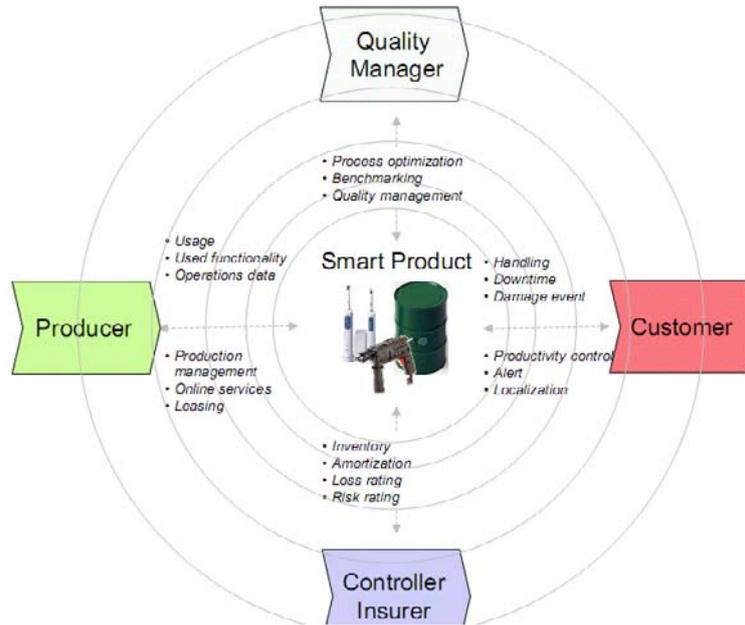
Die Entwicklung von RFID und ähnlichen Systemen kann gesehen werden als der Anfang des Internet der Dinge. Während bisher nur Computer im Sinne von PCs im Netz waren, können durch in Haushaltsgeräten eingebettete Systeme immer weitere Teile der materiellen Welt miteinander vernetzt werden, siehe Abbildung 4. Dementsprechend wächst auch die Komplexität der Verwaltung. Formale Systeme werden daher eine zentrale Rolle bei der Organisation und Vernetzung von intelligenten Geräten und Computersystemen spielen. Nur mit einer guten logischen Infrastruktur ist es auch möglich, die Informationen, die die unzähligen Systeme einspeisen werden, auch zu verarbeiten und damit zu nutzen. Dann werden auch kundennahe Prozesse optimiert werden können und allen Beteiligten immer weiteren finanziellen und qualitativen Gewinn beschere, mit der fortschreitenden Erfüllung der unterschiedlichsten Wünsche an den technologischen Fortschritt.

Literatur

- [Katoen] Christel Baier and Joost-Pieter Katoen. *Principles of Model Checking*. The MIT Press, 2008.
- [2] Graham Clark, Stephen Gilmore and Jane Hillston. *Specifying Performance Measures for PEPA*. Laboratory for Foundations of Computer Science, The University of Edinburgh, 2008.

UbiComp & Telos
 Prof. E. Fleisch
 April 6, 2005
 Page 27

Good products want to communicate.



Nutzungsbereiche der Netzanbindung von Haushaltsgeräten

Abbildung 4. Nutzungsgebiete von Ubiquitären Systemen

- [3] J. Hillston. *A Compositional Approach to Performance Modelling*. Cambridge University Press, 1996.
- [4] P. Liggesmeyer, M. Rothfelder, M. Rettelbach und T. Ackermann. *Qualitätssicherung Software-bastierter technischer Systeme*. Informatik Spektrum, 21(5):249-258, 1998.
- [5] Prof. E. Fleisch. *Ubiquitous Networks and Their Impact on the Traditional Telecommunication Industry*. Institute of Technology Management, University of St. Gallen (HSG). April 6, 2005.
- [6] Prof. Dr. Friedemann Mattern. Institut für Pervasive Computing, ETH Zürich. *Zukunftstrends für Smart Labels* Impulsreferat zum Workshop "Vernetzte Smart Labels". Feldafinger Kreis Symposium 2005
- [7] A. Harter and A. Hopper. A distributed location system for the active office. *IEEE Network Magazine*, 8(1):62-70, 1994.

Simultaneous Localization and Mapping

Thomas Lörcher

Universität Karlsruhe

1 SLAM

Das weithin bekannte Global Positioning System, kurz GPS, kann nach Wikipedia Deutschland[13] zwar mittlerweile durch eine verbesserte Infrastruktur in weiten Teilen der Erde eine relativ geringe Unschärfe von 0,01 bis 5 Metern erreichen, besitzt jedoch einige entscheidende Nachteile. Unter anderem ist das sogenannte Differential-GPS nur an Orten verfügbar, die Hilfsmesseinrichtungen besitzen. In anderen Gebieten kann lediglich eine Unsicherheit von zwischen 10 und 15 Metern gewährleistet werden. Außerdem durchdringen die Satellitensignale des GPS keine Wände oder Dächer, lassen sich also nicht in Gebäuden einsetzen. Deshalb sind andere Ansätze gefragt. Einer davon ist das sogenannte Simultaneous Localization and Mapping. Das mit SLAM abgekürzte Verfahren spielt im Bereich der Navigation mittlerweile eine zentrale Rolle. Dabei birgt es Schwierigkeiten und Komplikationen.

Beim Simultaneous Localization and Mapping wird zunächst von einem teilweise oder gar gänzlich unbekanntem Gebiet ausgegangen, das höchstens partiell eine absolute Positionierung erlaubt. Ein Agent beziehungsweise Roboter ermittelt in der Theorie zugleich die Karte des Gebiets, in dem er sich befindet, und seine eigene Position. Diese Anforderung kann in unzähligen Varianten erfüllt werden. Die später vorgestellten Varianten unterscheiden sich teilweise eminent voneinander.

1.1 Szenario Robotik

In der Robotik ist der Einsatz von SLAM in unterschiedlichsten Bereichen denkbar. Beispielsweise bei unbemannten Operationen unter Wasser [9] oder gar auf fremden Planeten, in denen sich der Roboter selbständig fortbewegen und unter Umständen an bereits befahrene Punkte zurückkehren soll, bietet sich ein solches Verfahren an. Gerade bei Unterwassereinsätzen, bei denen zusätzlich oftmals die Sicht eingeschränkt ist, setzt es neue Herausforderungen. Auch das Erstellen einer Karte spielt in unbekanntem Gebieten ohne Beisein eines Menschen eine zentrale Rolle. Unterwasserkartographie etwa setzt SLAM gezielt ein. Doch auch wesentlich populärere Roboter verwenden SLAM. Sowohl bei Saug- oder Putzrobotern für den Haushalt als auch bei automatischen Rasenmähern findet eine dynamische Kartierung und Navigation Anwendung. Solche Systeme verwenden aber meist einfacher zu implementierende Methoden. Beispielsweise das statische Programmieren von vorgegebenen Strecken.

1.2 Szenario Indoor

Einige innovative Versuche beschreiben den Einsatz von SLAM innerhalb von Gebäuden. Einsätze von Spezialeinheiten und vor allem Feuerwehmännern sollen durch gezielte Navigation in Gebäudekomplexen unterstützt werden. Gerade bei unbekanntem Terrain und einer eingeschränkten Sicht können beispielsweise Feuerwehmänner sicher zum Bestimmungsort und wieder zurück geführt werden, ohne bereits Vorabinformation des Gebäudeplans zu besitzen. Dabei spielt bei Einsätzen die Geschwindigkeit und die Zuverlässigkeit des Systems eine entscheidende Rolle. Schließlich geht es bei solchen Brandfällen um Minuten oder gar Sekunden. Und die Gefahr, Feuerwehmänner falsch zu leiten und damit schlimmstenfalls selbst ins Unglück zu schicken, sollte dabei auf ein Minimum reduziert werden.

2 Grundlagen

2.1 Absolute Positionierung

Um eine Kartierung und die damit verbundene Navigation zu ermöglichen, bieten sich zwei verschiedene Positionierungsarten an. Bei der absoluten Positionierung geht es darum, eine global eindeutige Koordinate zu erhalten. Die Ermittlung des Punktes, an dem man sich befindet, ist hier unabhängig von vorhergegangenen Navigationsbestimmungen. Der Ausdruck "global" bezieht sich hier auf ein abstraktes System. Die Erdoberfläche und damit die Weltkarte oder die Position im Bezug auf das gesamte Universum sind die gängigsten solcher Systeme.

Multilateration. Multilateration ist nach [16] ein System zur Bestimmung der Position eines Objektes. Dabei werden drei oder mehr weitere Instanzen benötigt. Die Instanzen sind derart miteinander synchronisiert, dass geringe Zeitunterschiede ermittelt werden können. Das Verfahren kann auf zwei Art und Weisen erfolgen.

1. Das Objekt stellt einen Sender dar, der in konstanten Intervallen Signale aussendet. Die aufeinander abgestimmten Instanzen sind in diesem Fall Empfänger, die mithilfe der **time difference of arrival**(TDOA) die Position des Objektes bestimmen können. Je mehr Empfänger vorhanden sind desto besser lassen sich Messfehler ausgleichen.
2. Das Objekt stellt einen Empfänger dar, der laufend in synchronisierten Intervallen Signale von den Instanzen, die Sender bilden, übermittelt bekommt. Diese Signale werden zur gleichen Zeit von den Instanzen abgesendet, damit das Objekt selbst seine Position im Bezug auf die Instanzen berechnen kann.

Mathematisch gesehen reduzieren bereits zwei Instanzen den potentiellen Positionsraum auf einen Hyperboloiden. Ein dritter und vierter Empfänger bzw. Sender ergibt nach der Theorie den exakten Standort des Senders und somit die eigene relative Position.

Leider wirken in der Praxis weitere Faktoren mit ein. Vor allem Fehler der TDOA-Messung begründen entscheidende Schwankungen der Ortung. Damit man dennoch eine robuste Annäherung erhält, werden verschiedene Optimierungs- und Annäherungsverfahren angewendet. Wie genau die Signale aussehen und mit welchem Medium sie versendet werden, spielt zunächst keine Rolle.

Multilateration wird oft fälschlicherweise mit ähnlichen Methoden wie Triangulation und Trilateration verwechselt.

2.2 Relative Positionierung

Im Gegensatz zur absoluten benötigt die relative Positionierung bereits vorangegangene Punkte, auf die sie sich bezieht. Sie wird meist durch Messung der Geschwindigkeit und Zeit, die im Bezug zu einem Anfangspunkt zurückgelegt wurden, bestimmt. Dabei muss der Bewegungswinkel auch berücksichtigt werden.

Inertial Navigation. Laut [15] ist ein Inertial-Navigationssystem eine Navigationshilfe, die mittels Rechnern und Bewegungssensoren ständig die Position, Geschwindigkeit und Orientierung eines sich bewegenden Objektes bestimmen kann. Zusätzlich können Barometer oder Magnetometer eingesetzt werden. Im Regelfall wird die Geschwindigkeit und die Position des zu unterstützenden Objekts durch Integration über die Winkeländerung und die Beschleunigung ermittelt. Einflüsse wie Unebenheiten, Magnetfelder oder veränderte Bodenverhältnisse beeinträchtigen die Beobachtungen stark. Methoden, diese einzudämmen, stellen häufig Filter dar. Speziell der erweiterte Kalmanfilter ist weit verbreitet. Ob diese Maßnahmen ausreichen, wird in dieser Arbeit untersucht.

Ultraschall. Neben Multilateration gibt es die Möglichkeit, über die **times of flight** (TOF) eine Entfernung zu ermitteln. Ultraschall hat die Eigenschaft, sich mit Schallgeschwindigkeit auszubreiten. Indem nun ein Radiosignal und gleichzeitig ein Ultraschallsignal absendet wird, kann die times of flight des Signals bestimmt werden. Radiosignale breiten sich nahezu mit Lichtgeschwindigkeit aus. Daher kann in normalen Szenarien (für Lichtgeschwindigkeit irrelevante Entfernungen) derart abstrahiert werden, dass von einem sofortigen Ankommen des Signals beim Empfänger ausgegangen werden kann. Mithilfe des Zeitunterschieds, der zwischen Ankunft des Radio- und des Ultraschallsignals liegt (annähernd times of flight), kann nun die Entfernung zueinander hergeleitet werden. Aber auch als Multilaterationsmedium lässt sich Ultraschall einsetzen.

Eine weitere wichtige Eigenschaft von Ultraschall ist, dass keine Wände durchdringt. Dieses Merkmal hat seine Vor- und Nachteile. Sie werden im weiteren Verlauf näher beleuchtet.

2.3 Diskrete Systeme

Lokalisierung kann auf verschiedene Wege repräsentiert werden. Bisher ist davon ausgegangen worden, eine feingranulare Position zu erhalten. Damit sind im

Normalfall konkrete Koordinaten gemeint. Dies ist aber nicht immer nötig. Oft genügt es, den Bereich diskret einzuschränken, das heißt das zu untersuchende Gebiet in unterschiedliche Bereiche aufzuteilen. Dem Objekt ist es dabei lediglich möglich, eine sehr grobe Ortsangabe zu erhalten. Das gängigste diskrete System ist die Aufteilung in Zimmer oder Räume.

Gerade Medien, die Wände nicht oder zumindest nur mit geringer Stärke durchdringen, sind für ein diskretes System angemessen. Darunter befinden sich unter anderem Ultraschall und Infrarot.

RFID. Laut [17] ermöglicht Radio Frequency Identification die “automatische Identifizierung und Lokalisierung von Gegenständen und Lebewesen[.]”. Ein RFID-Chip ist ein Transponder, der im Regelfall keine eigene Energieversorgung besitzt. Ein Lesegerät “erzeugt ein elektromagnetisches Hochfrequenzfeld”, das den Transponder mit Energie füttert. Dieser antwortet mittels elektromagnetischer Wellen und mit seiner eigenen ID darauf. Die geringe Reichweite der Chips stellt nicht immer einen Nachteil dar.

RFID kann teilweise selbst Energieträger sein. Da das System aber gerade durch die passive Energienutzung sehr populär geworden ist, werden aktive RFID-Tags nur selten verwendet.

Infrarot-Beacons. Wie bei Ultraschall durchdringt Infrarot keine Wand. Daher kann man Infrarot-Beacons ebenfalls zur diskreten Ortsbestimmung verwenden. Gegenüber Ultraschall ist Infrarot relativ kostengünstig. Außerdem sind Infrarot Transmitter kleiner und haben eine geringere Signalprozessanforderung[1]. Daher werden sie gerne in ubiquitären Systemen wie Sensornetzwerken oder ähnlichem eingesetzt. Ein entscheidender Nachteil von IR-Beacons ist die wesentlich schwerer zu erreichende Entfernungsbestimmung. Dadurch, dass Infrarot in etwa dieselbe Ausbreitungsgeschwindigkeit wie Radiowellen hat, ist an eine time of flight Abstandsmessung nicht zu denken. Infrarotbeacons werden in der Regel immer den nächst gelegenen (den am schnellsten antwortenden) Sendern zugeordnet und erhalten daher nur eine grobgranulare Position.

2.4 Probabilistische Filter

Um simultaneous localization and mapping zu ermöglichen, werden in diskreten Zeitabständen Messungen durchgeführt. Diese können Entfernungs-, Winkel oder andere Messungen sein. Da in der Praxis relativ große Fehler auftreten, müssen die Messungen ständig angepasst werden. Sogenannte Filter bieten eine Alternative um die Abweichungen zu relativieren. Teilweise können selbst Ausreißer neutralisiert werden.

Extended Kalman Filter. Der in diskreten Systemen am häufigsten verwendete Filter nennt sich Erweiterter Kalman Filter. Dieser stellt die Umgebung in einem Statusvektor dar. Typische Komponenten des Vektors sind etwa die

eigene Position, der eigene relative Winkel(im Bezug auf eine vordefinierte Ausrichtung) und die Positionen anderer Landmarken oder Beacons.

Typischer Statusvektor:

$q_k = [x_k, y_k, \theta_k, x_{1k}, y_{1k}, x_{2k}, y_{2k}, \dots]$, wobei $y_{1k}, x_{2k}, y_{2k}, \dots$ die Positionen der Beacons und x_k, y_k, θ_k Position und Ausrichtung des Roboters darstellen.

Der Statusvektor wird oft auch schlicht Zustand genannt. Der Kalman-Filter schätzt mithilfe des im letzten Zeitschritt ermittelten Zustands einen neuen apriori Wert. Dieser wird gewichtet und mit der neu gemessenen Umgebung zum aposteriori Wert verschmolzen. Eine solche Verschmelzung wird auch posteriori-Schätzung genannt und bietet eine fehlertolerante Positionsbestimmung. Mathematisch gesehen wird eine sogenannte Taylornäherung erster Ordnung verwendet. Ein Taylorpolynom stellt nach [14] eine Annäherung einer Funktion um einen speziellen Entwicklungspunkt dar.

Im folgenden wird das n-te Taylorpolynom von f dargestellt.

$$T_n(x) := f(a) + \left(\frac{f'(a)}{1!}\right)(x-a) + \dots + \left(\frac{f^{(n)}(a)}{n!}\right)(x-a)^n$$

Es wird also die Funktion, die zugrunde liegt, zum Zeitpunkt a geschätzt. Je mehr "Erfahrung" in Form von bekannten Punkten der Kurve gegeben sind, desto mehr Differenzgleichungen können aufgestellt werden und somit desto mehr Ableitungen können die Schätzung nähren.

Beim EKF wird lediglich das Taylorpolynom des ersten Grades verwendet, was nur eine leichte Verbesserung der Messung mit sich zieht. Dafür ist nur der letzte Roboterzustand nötig. Dies erleichtert jedoch die Implementierung und Analyse der Schätzer.

Partikelfilter. Der Partikelfilter oder der im Bezug auf Lokalisation sogenannte Monte Carlo Lokalisationsansatz arbeitet mit Partikeln. Das Paper [5] erläutert ihn folgendermaßen. Ein diskreter Zustand hat eine bestimmte Anzahl von Partikeln, die ein Tupel $\langle\langle x, y, \alpha \rangle, p\rangle$ aus Roboterstatus, also Position und Winkel des Roboters, und eine daran geknüpfte Wahrscheinlichkeit beschreibt. Die Wahrscheinlichkeit aller Tupel ergibt zusammen 1. Zunächst sind alle Partikel im Nullpunkt und besitzen dieselbe Wahrscheinlichkeit. Sobald sich der Roboter bewegt, werden N neue Partikel erzeugt. Dabei wird auf den bisherigen Tupelpool zurückgegriffen. Je größer die Wahrscheinlichkeit eines Tupels, desto eher wird er weiterverwendet. Da die Odometry nicht fehlerfrei arbeitet, wird die Bewegung probabilistisch mit $P(l|l', a)$ dargestellt. Dieser Term repräsentiert dabei die Wahrscheinlichkeit, dass der Roboterstatus l bei Bewegung a nach vorangegangenem geschätzten Status l' eintritt. Dabei wird angenommen, dass die Menge der Wahrscheinlichkeiten normalverteilt ist. Einer dieser potentiellen neuen Stadien wird je zufällig mit Wahrscheinlichkeit $P(l|l', a)$ ausgewählt und in die nächste Partikelmenge eingefügt.

Im nächsten Schritt wirken sich dann die Sensormessungen auf den Filter aus. Die Partikelwahrscheinlichkeiten werden jetzt mit

$$p \leftarrow \alpha P(s|l)$$

neu gesetzt. Dabei bezeichnet s die Messungen und α eine Normierungskonstante (um die Summe aller p auf eins zu halten). Interpretiert bedeutet dies, je früher die Beobachtung mit dem neuen Partikel übereinstimmt, desto eher wird er später weiterverwendet.

Partikelfilter sind relativ leicht zu implementieren und benötigen wenig Rechenressourcen. Desweiteren haben sie durch N eine konstante Größe und sind daher meist sehr schnell. Für mehr als 3 Dimensionen wächst der Aufwand jedoch exponentiell.

3 Navigation ohne SLAM

Szenarien, in denen eine Navigation ohne bereits existierende Karten erfordert wird, häufen sich in der modernen Welt immer mehr. Ubiquitäre Systeme, die Menschen in unterschiedlichsten Lagen unterstützen sollen, oder autonome Roboter, werden zunehmend mit einem dynamischen Mappingszenario konfrontiert. Um eine Positionierung in unbekanntem Umgebungen gewährleisten zu können, wurden in den letzten Jahrzehnten viele Ansätze verfolgt. Einer der grundlegenden ist dabei der Einsatz eines reinen Inertialsystems.

Anwendung einer Inertiallösung.

Die Arbeit [4] stellt ein System vor, das aus einem dreiachsigen Beschleunigungssensor, einem Gyroskop und einem Magnetometer besteht. Diese Einheit wird an einem Schuh befestigt, um den Träger so wenig wie möglich zu behindern. Um aus den Beinahmessungen eine brauchbare Ausgabe zu erzeugen, wird der Bewegungsvorgang beim Laufen näher unter die Lupe genommen. In dem Paper wird das Gehen in zwei Phasen unterteilt. Die "stance" Phase und die "swing" Phase. In der erstgenannten Phase berührt der Fuß den Boden. Die Geschwindigkeit wird zurückgesetzt und auf Null gehalten, während in der Swingphase die Beschleunigung des Sensors doppelt integriert wird. Um die Stancephase auszumachen, wird ein Schwellenwert festgelegt, den das Produkt der Norm der Beschleunigung und der Norm der Drehrate überschreiten muss. Bleibt dieses Produkt 0.2 Sekunden unter dem Schwellenwert, wird eine Stancephase angenommen. Die Geschwindigkeit und vom Gyroskop gemessene Drehbewegung des Probanden wird mit der vergangenen Zeit verrechnet, um die zurückgelegte Strecke zu ermitteln.

Der dazu durchgeführte Versuch wurde mit einer Samplefrequenz von 100 Hz sowohl bei der Winkel- als auch der Beschleunigungsmessung durchgeführt. Die Algorithmik wurde in ein Sony Vaio Handheld verlagert. Diese wertete die Sensordaten aus und berechnete daraus den zurückgelegten Bahnverlauf. Es wurden verschiedene Wegeszenarien durchlaufen um ein möglichst allgemeines Ergebnis zu erzielen. Darunter eine einfache, gerade Strecke über 88m, eine L-förmige Strecke über 54 Meter, eine 11,5 Meter lange rechtwinkelige Strecke und schließlich

einen Korridor, bei dem mehrere Räume betreten und wieder verlassen wurden.

Carl Fischer et al. ermittelten die Fehlerrate, die bei diesem Versuch auftraten. Bereits bei der ersten Strecke wurde ein Fehler von 2% der insgesamt zurückgebrachten Strecke verzeichnet. Die L-förmige Strecke wies mit 8% ein erheblich größeres Fehlverhalten auf. Auch der rechtwinkelige Bahnverlauf lieferte mit 7% ein unzureichendes Ergebnis. Wie man erwarten durfte, traten die größten Orientierungsfehler nach 180°-Drehungen auf. Der Korridor brachte allerdings den größten Fehler. Der real zurückgelegte Weg deckte sich kaum mit dem ermittelten.

Als zusätzlicher Test wurde eine Bahn aufgebaut, bei der starke magnetische Störungen die Messungen des Magnetometers beeinflussten. Das Ergebnis war eine völlig unzureichend genau beobachtete Bahn.

Dieser Ansatz eines Bewegungssensorsystems macht deutlich, wie ungenau und unzuverlässig daraus gewonnene Ergebnisse sind. Bereits nach wenigen Metern treten Fehler der Positionsbestimmung auf. Im Verlauf der Zeit summiert sich die Ungenauigkeit derart, dass bereits nach wenigen zig Metern und vor allem nach oftmaligem Drehen ein unbrauchbares Ergebnis geliefert wird. Viele Versuche, das Inertialsystem zu perfektionieren, haben zwar zu sichtlich verbesserten Ergebnissen geführt, erliegen aber früher oder später trotzdem ihrer Positionsunschärfe.

4 SLAM in der Robotik

Navigation in neuen, unbekanntem und eventuell unübersichtlichen Gebieten ist eine entscheidende Voraussetzung, um autonome Roboter zu verwirklichen. Um selbstständig eine Karte zu erzeugen und mit dieser Aufgaben zu erfüllen, wird eine absolute oder relative Positionierungstechnik vorausgesetzt.

4.1 Optisches SLAM

In der Arbeit [3] wird ein System beschrieben, das sich mithilfe einer "Panasonic WV-BL202"-Kamera im ersten Versuch an einer einzigen und später an zwei Landmarken orientiert. Die Anforderung dafür ist der Winkel, in dem sich die Landmarke relativ gesehen zum Roboter befindet. Deshalb ist die absolute Peilung des Roboters vonnöten. Dazu verwendet der Roboter einen Kompass. Die Landmarken werden durch kleine "Maglites" mit hoher Intensität repräsentiert. Der verwendete Kalmanfilter wurde zunächst mit dem Startwert $(x,y) = (0,0)$ gefüttert. Mithilfe der Frames wurden Peilungsschätzungen erzeugt, die mit dem apriori Wert des letzten Zyklus des Kalmanfilters zur aposteriori Schätzung vereint wurden. Dazu wurden in zwölf Sekunden fünf Frames erzeugt.

Daniel Boley et al. erörtern hier einen verbesserten Ansatz, der eine Alternative zum Kalmanfilter darstellen soll. Sie verwenden ein überbestimmtes lineares System der Form $Ax = b$. Die Matrix A besteht dabei aus den Zeilen

$a_i^T = (\sin(\alpha_i) - \cos(\alpha_i))$ und dem Ergebnisvektor
 $b_i = -\sin(\alpha_i) \cdot d_i$,
 wobei α_i den gemessenen Winkel zur Landmarke darstellt.

Allerdings wird im Paper nur nebenbei erwähnt, dass die fortbewegte Distanz d_i für die weitere Berechnung erforderlich ist. Ein ungeliebter Umstand, da weitere Komplexität in den Versuch einfließt. In der Veröffentlichung nutzen die Autoren nicht etwa ein Inertialsystem, sondern gehen von einem gleichmäßig fahrenden Roboter aus, der sich relativ genau mit 20mm pro Sekunde bewegt. Eine fatale Annahme, die zu starken Fehleinflüssen der Berechnung führt. Lediglich die aktuellsten n Einträge werden in der Matrix vermerkt. Da bei der Messung starke Schwankungen auftreten, kann das lineare System nicht einfach gelöst werden. Daher muss das System auf das "kleinste Quadrate"-Problem abgebildet werden. Dabei soll die Größe

$$\min_x \| r \| = \min_x \| Ax - b \|_2$$

minimiert werden. Diese Art der Minimierung stellt eine Regressionsgerade der Gleichungen des Systems dar. Mathematische Verfahren lösen diese Gleichungen und bieten eine gute Annäherung der Karte und der Position des Roboters.

Ein entscheidender Nachteil des Ansatzes waren naive Annahmen über die Bewegung. Für das Szenario "Optisches SLAM" wäre ein Inertialsystem notwendig gewesen. Außerdem sind optische Systeme sehr störungsempfindlich. Zum einen erschweren es schlechte Lichtverhältnisse, ein Objekt zu erkennen, und zum anderen liefern Kameras nicht immer optimale Bilder. Und aus diesen immer genauere Winkel zu extrahieren, stellt eine weitere Hürde dar.

Ein zentrales Argument gegen die Nutzung dieser Art von optischem SLAM ist die mangelnde Skalierbarkeit. Die angedeutete Matrix bezieht sich keinesfalls auf die gesamte Karte. Sie verbindet den Roboter lediglich mit einer Landmarke. Bei immer größer werdenden Karten nimmt die Matrix ungeahnte Ausmaße an. Dieser frühe Ansatz sollte also nur ein Zwischenschritt in Richtung Lösung des Mappingproblems darstellen.

4.2 Distributed Particle-SLAM

[6] beschreibt eine fortgeschrittenere SLAM Variante. Ein mit einem Laserrangefinder ausgestatteter Roboter verwendet einen abgewandelten Partikelfilter, der eine begrenzt große Karte repräsentiert. Diese hat ein fest vorgegebenes Raster von Belegungen, bildet also ein Belegungsgitter. Der Laserrangefinder kann durch Aussenden von Laserstrahlen die times of flight der zurückkehrenden Strahlen ermitteln und damit die Entfernung zu Hindernissen oder Wänden bestimmen.

Im naiven Ansatz des Partikelfilters würde jedes Sample eine gesamte Karte von

Hindernissen und den Zustand des Roboters beinhalten. Da aber bei der Erzeugung neuer Partikel immer wieder die größtenteils gleichgebliebene Karte im Speicher kopiert werden müsste, wird in diesem Paper eine neue Richtung eingeschlagen. Austin Eliazar et al. verwenden statt vieler aufgeblähter Partikel eine global einmalige Karte, in der die Samples repräsentiert werden. Jeder Sektor des Raster enthält einen balancierten Baum von Partikel-IDs. Sobald ein Partikel eine Beobachtung macht, wird diese in dem Baum des betroffenen Sektors vermerkt. Um den Wert eines solchen Bereichs zu ermitteln, durchsucht das Partikel sämtliche seiner Vorfahren nach einem Eintrag. Ist keiner dazu vorhanden, wird die Belegung als unbekannt angenommen. Damit weder der Baum noch die Vorfahren-/Nachkommendatenstruktur zu groß wird, werden nach optimierten Verfahren unbrauchbare Einträge (z.B. Vorfahreinträge ohne "Messfortschritte") aus den Datenstrukturen entfernt.

Der Versuch dazu verwendet einen iRobot ATRV Jr., der mit einem Laserrangefinder ausgestattet ist. Dieser ist 7cm über dem Boden angebracht und kann in einem Bereich von 180° Messungen ausführen. Dabei hat er eine effektive Messweite von 8m und typischerweise einen Fehler von unter 5mm. Das Odometriemodell hat hingegen eine Standardabweichung von 25% bei seitlichen Bewegungen und 50% bei Drehungen. Der Test wird in einem Korridor der Größe 16m auf 14m getätigt. Die Karte hat eine Rastergröße von 3cm auf 3cm. Die Anzahl der Partikel beläuft sich auf 9000. Der Roboter fährt dabei gegen den Uhrzeigersinn im Kreis zurück zu seinem Ausgangspunkt.

Ein kompletter Rundgang führt den Roboter an den genauen Ausgangspunkt seiner Reise. Demnach repräsentiert er damit ein gutes Ergebnis des Versuchs. Ein entscheidender Punkt bleibt aber zu klären: Je nach Anzahl von Partikeln verändert der Proband seine Genauigkeit. Wieviel Partikel sind tatsächlich für ein zuverlässiges System notwendig?

Fazit: Ein schöner Artikel und Versuch erzielt eine gute Präzision, hat jedoch den Nachteil, ein festes, vorgegeben großes Raster und die damit verbundene mit vordefiniertem Maß angelegte Karte zu benötigen. Das Effizienz- Genauigkeitsverhältnis spricht hingegen für DP-SLAM.

5 SLAM für Ubicomp Szenarien

Um sich in unbekanntem Umgebungen orientieren zu können, bedurfte es bisher immer optimalen Bedingungen. Auffällige Gegenstände oder Landmarken wurden als Anhaltspunkt zur Navigation verwendet. Sobald aber problematischere Verhältnisse vorherrschten, mussten skurrile Vorgehensweisen eingeführt werden, um die Auswirkungen der schlechten Lage einzuschränken. Ein typisches Szenario stellt etwa der Einsatz der Feuerwehr in Brandgebäuden dar. Teilweise mit Sichtweiten unter einem Meter und sich verändernden Umgebungen zum Beispiel durch einfallende Räume oder Türen erschweren es den Rettern, Opfer zu finden und zu bergen. Die Verantwortlichen setzen oft die sogenannte Feuerwehrleine ein. Ein spezielles Seil, das ein erstes Team durch die zu durchschreitenden Räume trägt, ermöglicht es, ihm zu folgen und wieder zurück zu

finden. Doch das System hat entscheidende Nachteile. Das Seil macht es extrem umständlich, kurzfristige Plan- oder Richtungsänderungen auszuführen. Außerdem machen es zwischenzeitlich veränderte Umgebungsverhältnisse schwierig, wieder zurückzufinden.

5.1 Einsatz von RFID-Chips

Ein etwas aktuelleres Unterfangen erzeugt eine diskrete Lokalisierung, die die Zuordnung eines Probanden zu einem Raum oder genauer zu einer Tür liefert. Dieses System verwendet RFID-Tags.

Im Paper [11] wird ebenfalls ein Trägheitssystem verwendet. Hier wird es "Micro-Electro-Mechanical System", kurz MEMS, genannt. Es besteht wieder aus Beschleunigungssensoren, Gyrometern, Magnetometer und Barometer. Das Barometer dient dazu, die Etage auszumachen oder einen Treppenanstieg zu rekonstruieren. Die Messeinrichtungen werden an Ober- und Unterschenkel sowie auf der Brust angebracht. Die Daten werden in einem Takt von 50 Hz gesammelt und dann mit einem "Adaptive Extended Kalman Filter" geglättet. Der Filter, der eine leichte Abwandlung des EKF darstellt, relativiert somit leichte Fehler oder gar Ausreißer.

Hier geht es ebenfalls um ein System zur Unterstützung von Feuerwehrmännern, die mit Hilfe des Navigationssystems zur Unfallstelle und von dort zurück geleitet werden sollen. Gerade bei starker Sichtbeeinträchtigung soll eine Rettung von Opfern ermöglicht werden. Bedingung hierfür ist aber ein vorhandener Plan des Gebäudekomplexes. Der Anwendungsfall wird in zwei Phasen unterteilt. In der ersten Phase schwärmt ein erstes Team von Feuerwehrmännern aus und bringt in der Mitte von Türpfosten solche RFID-Tags an. Parallel berechnet das Inertialsystem bzw. MEMS die etwaige Positionsveränderung. Sobald ein Chip angebracht ist, überprüft ein verantwortlicher Feuerwehrmann, welche Tür des Plans am ehesten zu der vom Inertialsystem des Teams ermittelte Position passt. Eine Datenbank speichert die ID in Verbindung mit der Tür und ihren tatsächlichen Koordinaten. Sobald ein RFID-Tag gespeichert ist, werden alle Inertialsysteme der Feuerwehrmänner, die sich in unmittelbarer Sendenähe befinden, auf die absolute Position des Chips zurückgesetzt. Dies erlaubt es den Systemen, die ständig produzierten Fehler auf ein Minimum zu reduzieren. Ist eine Tür oder ein dahinter gelegener Raum unpassierbar, werden die Tags entfernt, sodass diese nicht mehr fälschlicherweise begangen werden. Abkürzungen und Umleitungen um Gefahrenstellen können dann in Kombination der Karte und den Chips gefunden werden.

Um das System zu testen, wurde auf dem Campus der "Ecole Polytechnique Fédéral de Lausanne" in der Schweiz ein typisches Einsatzzenario erforscht. Ein Beauftragter betritt den Campus am Haupteingang und läuft durch den Korridor um ein Büro zu erreichen. Dort ist ein Feuer ausgebrochen. Einige

Menschen warten darin auf eine Rettung. Auf dem Weg zum Ziel bringt der Agent an den passierterten Türen RFID-Tags an. Seine Position wird durch das Inertialsystem ermittelt und durch die Tags immer wieder korrigiert. In den konkret ausgeführten Testreihen wurde sowohl das reine MEMS Ergebnis als auch das Resultat in Kombination mit den RFID-Chips berechnet. Vier mal wurde der Versuch durchlaufen. Zwar war die Karte und damit das Szenario prinzipiell immer dasselbe, doch die Ergebnisse lügen nicht: In jedem einzelnen Durchgang war die Hybridlösung deutlich genauer. Jeder RFID-Chip konnte der richtigen Tür zugeordnet werden, wobei die reine Inertiallösung zu immer größerer Ungenauigkeit tendierte.

Als Ergebnis stellen die Autoren fest, dass 90% der Fehler unter fünf Metern liegen. Hingegen ist bei dem alleinigen Inertialsystem ein Durchschnitt der Abweichung von 19,3 Metern ermittelt worden. Die Standardabweichung von 9,8 Metern zeigt die starke Streuung des Fehlers um den Mittelwert.

Als Fazit kann man sicherlich nennen, dass das System im Test sehr genau und zuverlässig reagiert hat. Leider wurde es nur unzureichend getestet und kann daher nicht als bewiesen robust angesehen werden. Allerdings scheint es der Vernunft Genüge getan um als hinreichend präzise zu gelten. Doch die Idee dieser RFID-Chips hat einen entscheidenden Nachteil. Um eine Navigation ermöglichen zu können, muss eine detaillierte Karte vorhanden sein. Um im Notfall möglichst unbürokratisch und direkt helfen zu können, darf diese Forderung nicht existieren. Besitzer des Papiers könnten unabkömmlich sein. Außerdem existieren nicht für alle Gebäude Pläne. Und um eine sofortige Rettung zu gewährleisten, darf der Zeitraub der Kartenbeschaffung nicht in die Rettungsaktion einfließen. Desweiteren muss ein zentrales System ständig abrufbar sein, damit jederzeit auf die Daten der Datenbank zugegriffen werden kann.

5.2 Einsatz von ersten kommunizierenden Beacons

Active Badge. Damit man die Entwicklung im Gebiet der Lokation weiter nachvollziehen kann, ist ein Sprung in die Vergangenheit nötig. Eigentlich war das Ziel dieses Papers von starrer Infrastruktur weg zu kommen, um auch in unvorbereiteten Umgebungen Navigation zu gewährleisten. Der anschließend präsentierte Aufbau benötigt zwar gerade eine solche Voraussetzung, enthält jedoch ein grundlegendes Konzept, das weiter untersucht werden soll.

Bereits 1992 entstand jenes System, das die Ortung innerhalb von Gebäuden erlaubte. In der Arbeit [12] wird der Aufbau von Active Badge beschrieben. Ausgestattet mit einem Netzwerk von Sensoren, das an eine "Master Station" angeschlossen ist, kann es sogenannte Badges erkennen und lokalisieren. Dazu müssen die Badges, die beispielsweise Personen an ihrer Kleidung tragen, alle 15 Sekunden ein Signal senden. Über Infrarot werden somit die eindeutig vergebenen IDs der Träger an die Sender und somit an den Master weitergegeben. Dieser ermittelt anhand der erhaltenen ID und des Sensors, in welchem Raum oder in welchem Segment sich das Badge befindet.

Für Active Badge muss allen Sensoren ein Netzwerk, an das auch der Master angeschlossen ist, zur Verfügung stehen. Im allgemeinen ist das System grobgranular und ermöglicht lediglich eine Zuordnung des Beaconträgers zu einem Raum. Active Badge gewährleistet also eine diskrete Positionsbestimmung.

Active Bat. Um den möglichen Positionsraum einzuschränken, war es notwendig, nach Alternativen zu suchen. Die Entdeckung von Ultraschalltransmittern und -empfängern eröffnete die Möglichkeit, Abstände zwischen Sensoren und Bats herzuleiten. Active Bat wird in [2] beschrieben.

Das Prinzip der Entfernungsmessung baut auf dem in den Grundlagen erläuterten time-of-flight-Ansatz auf.

Durch Reflektionen entstehen nur dann Störungen, wenn sich zwischen Sender und Empfänger ein Hindernis befindet. Denn das Ultraschallsignal, das zuerst eintrifft, folgt dem kürzesten Pfad und repräsentiert somit den direkten Weg. Das Radiosignal durchquert problemlos Wände und Mauern und wird von derartigen Effekten nicht beeinflusst.

Angenommen zwischen Sender und Empfänger befindet sich kein Hindernis, so kann der Master, der in diesem Szenario ebenfalls über ein Netzwerk mit allen Sensoren verbunden ist, die Entfernungen des Bats zu verschiedenen Sensoren ermitteln und seine Position durch Multilateration (siehe Grundlagen) auf einen kleinen Bereich einschränken. Die in einem Gitter angebrachten und damit an vorgesehenen Positionen installierten Sensoren erlauben eine Lokalisierung von Bats innerhalb von 9cm um die tatsächliche Position bei 95% der Messungen.

5.3 Einsatz von Wireless Sensor Networks

Zwar kann man von keinem bereits vorinstallierten Netzwerk von Beacons in unbekanntem Terrain ausgehen, doch könnte man ein solches dynamisch aufbauen. Natürlich bietet es sich in Notfällen nicht an, Landmarken auszulegen und diese mithilfe von Netzkabeln zu verbinden. Die Lösung dafür sind kabellose Sensornetzwerke.

SLAM mithilfe von konvexen Schranken. Die Lösung von [10] geht zunächst von einem vorgegebenen Sensornetzwerk aus, bei dem alle Knoten in der Lage sind, Entfernungen zu messen, diese Werte zu sammeln und auszuwerten, und an eine Senke weiter zu versenden. Mithilfe der daraus gewonnenen Erkenntnisse können dann Roboter eingesetzt werden um gezielt Beacons bzw. Sensoren im unbekanntem Gebiet auszulegen. Es wird mit einem Netzwerk gerechnet, deren Knoten eine Konnektivität von teilweise eins oder unwesentlich höher besitzen. Um dennoch eine Karte erstellen zu können, muss ein mobiler Knoten eingeführt werden. Dieser fährt eine vordefinierte Strecke ab und wertet dabei die Entfernungsmessungen zu den einzelnen Beacons aus. Der wesentlich rechenstärkere dynamische mobile Knoten geht im voraus von einem Messfehler aus. Dieser wird bei der Entfernungsbestimmung zu einem einzelnen Sensor mithilfe einer

oberen und unteren Fehlerschranke mit in die weitere Rechnung aufgenommen. Daraus erhält er quadratische Constraints, die die zum mobilen Knoten relative Position einengen kann. Es ergeben sich konvexe und konkave Constraints:

$$\begin{aligned} v_{ij}^T v_{ij} &= x^T (P_j - P_i)^T (P_j - P_i) x \leq (r_{ij}^+)^2 \\ v_{ij}^T v_{ij} &= x^T (P_j - P_i)^T (P_j - P_i) x \geq (r_{ij}^-)^2 \end{aligned}$$

mit x als Zustandsvektor des eingesetzten Kalmanfilters, P_i, P_j die Matrizen, die aus x die Position der beiden Knoten i und j extrahiert und r_{ij}^+, r_{ij}^- die obere bzw. untere Messschranke und v_{ij} der Abstand zwischen den Knoten i und j . Nach jedem Schritt werden wieder Kalmanfilterungen ausgeführt, die den Zustand x des Raums glätten. Außerdem werden die von den Constraints gebildeten Polytope im Konfigurationsraum (nicht 3D-Karte sondern Raum bezüglich v_{ij}) durch iterative Messungen immer kleiner und erzeugen daher ein genaueres Abbild der Realität. Der Schwerpunkt des Polytops im Konfigurationsraum wird als eingesetzte Entfernung zur Berechnung der Positionen im Zustandsvektor x herangezogen.

Im Versuch wird eine Simulation eingesetzt. Da aber auch diese Fehler der Bewegung des mobilen Knoten produziert, werden auch diesbezüglich Constraints eingebaut. Es werden bloß konvexe Constraints verwendet. Beim virtuellen Roboter wird pro Zeitschritt von einem Geschwindigkeitsfehler von $\delta v = \pm 0.05v$ und einem Richtungsfehler von $\delta\theta = \pm 5^\circ$ ausgegangen. Der Roboter fährt in einem Umfeld, in dem 9 Knoten ausgebracht sind, eine Acht nach. Einmal werden zufällig Messfehler von bis zu ± 0.2 mal der echten Entfernung produziert, ein zweites Mal werden gezielt in jeder vierten Messung Ausreißer von zwischen 50 und 100% eingebaut.

Im ersten Versuch wurden respektable Positionsbestimmung sowohl des Roboters als auch der Knoten bestimmt. Die tatsächlichen Objekte lagen alle in den eingeschränkten Bereichen, die allerdings noch eine nicht unwesentliche Größe aufwiesen.

Im zweiten Versuch ergab sich ein um einiges schlechteres Abbild der Umgebung. Die Unsicherheitsregionen vergrößerten sich dabei durchschnittlich zu denen, die ohne Ausreißer ermittelt wurden, um etwa 10%.

Ein weiterer Versuch testete die Integration von konkaven Einschränkungen, die aber nur zu unwesentlich besseren Ergebnissen führte. Schlussendlich stellt dieser Versuch einen guten Ansatz dar, wird aber durch Bewegungs- und Messfehler keine ausreichende Genauigkeit für feinere Navigation ermöglichen. Außerdem ist er nicht genügend getestet worden. Reine Simulationen geben selten die Realität genau wider.

SLAM mit Virtuellen Knoten Mit immer schnelleren, kleineren und energiesparenderen Prozessoren und Recheneinheiten können Sensornetzwerke mittlerweile wesentlich aufwändiger gestaltet werden. Beacons, die sowohl Sender als auch Empfänger sein können und eine ansehnliche Verarbeitungsgeschwin-

digkeit besitzen, können dabei den sich fortbewegenden Agenten noch gezielter unterstützen. [7] beschreibt unter anderem einen solchen Ansatz.

Hier sind die Beacons sehr mächtig. Sie bilden ein vollständig verbundenes Netzwerk, bei dem jeder Knoten Informationen über die gesamte, bisher ermittelte Karte hat. Der Agent kann sodann Informationen von jedem erreichbaren Knoten über die Umgebung abrufen. Mit dieser neuen Herangehensweise verschiebt sich das SLAM-Problem etwas. Schließlich muss der Agent nicht selbst zugleich seine sowie die Position der Beacons ermitteln, sondern verlagert einen großen Teil der Komplexität des Problems in ein Sensornetzwerk. Um mathematisch eine eindeutige Karte zu erhalten, muss ein Sensornetzwerk eine Konnektivität von wenigstens drei oder vier besitzen. Per Multilateration könnten die Entfernungen dann zu einem Abbild der Umgebung fusioniert werden. Da diese Voraussetzung in diesem Szenario nicht gegeben ist, erzeugt der Agent zu diskreten Zeitpunkten sogenannte virtuelle Knoten. Dazu bewegt er sich an einen Punkt, an dem er Kontakt zu einigen Knoten hat und erzeugt Entfernungsmessungen.

Die Beacons, die in geregelten Abständen aktuelle Abstandsmessungen zu ihren Nachbarn machen, können den virtuellen Knoten wie einen tatsächlichen Knoten in ihrer internen Repräsentation verwenden. Mit dieser Methode kann der Agent gezielt Cluster erzeugen, um damit eine größere Konnektivität zu erzielen. Als Resultat kann das Netzwerk eine Karte der Umgebung erzeugen und somit die Position des Agenten bestimmen.

Im Versuch werden Mikrokontroller, die jeweils einen Radiotranceiver und vier Paar Ultraschallsensoren besitzen, als Beacons verwendet. Um eine Entfernung zu messen, wird wieder die times of flight Methode herangezogen. Um zu verhindern, dass Reflektionen die Messungen beeinflussen, wird lediglich das Ultraschallsignal ausgewertet, das zuerst eine der Sensoren erreicht. Die darauf folgenden werden ignoriert. Innerhalb des Netzwerkes sind die Knoten außerdem derart synchronisiert, dass zu einem Zeitpunkt höchstens ein Knoten ein Radiowellen-Ultraschallpaar abschickt. Dafür ist ein spezielles Zeitscheduling implementiert worden.

Das Experiment zeigte, dass das System sehr stabil und genau arbeitete. Von 20 014 Entfernungsmessungen wurden lediglich 6 Fehlbestimmungen verzeichnet. Die Standardabweichung der Messungen betrug weniger als 0,04 Meter, wobei die meisten Fehler unter 0,05 Meter erzeugten. Das Ergebnis der Testreihe waren beinahe perfekt bestimmte Knotenpositionen. Der Bahnverlauf hingegen zeigte im Vergleich zu anderen Verfahren jedoch Verbesserungspotential.

Zusammenfassend kann man behaupten, dass die Vorgehensweise klare Vorteile zu seinen Vorgängern aufweist. Zum einen wird hier ein dezentrales System vorgestellt, das eine gute Positionierung erlaubt. Außerdem ist ein Netzwerk, das sich ständig aktuell hält und zusätzlich Informationen von einem mobilen Knoten einholt, sehr robust. Denn Messungen wiederholen sich ständig und ermöglichen dadurch eine korrigierende Redundanz der Lokalisierung. Allerdings hat es den Nachteil, dass falsche Messungen virtueller Knoten nicht verbessert

werden können. Dies führt unter Umständen zu weiterer Ungenauigkeit. Der kritischste Punkt jedoch ist die Frage, wie skalierbar ein solches System und dessen zugrundeliegendes Schedulingverfahren tatsächlich sind. Für kleinere Einsätze, gerade wie das Feuerwehrszenario dürfte diese Variante jedoch eine Alternative darstellen.

6 Andere Medien

6.1 Subsea-SLAM

Um zu demonstrieren, dass SLAM nicht nur auf Robotik und Indooranwendungen beschränkt ist, wird hier noch eine andere Anwendung [9] beschrieben. Es handelt sich um ein System zur Navigation eines autonomen Unterwasserfahrzeugs. Mittels am Meeresboden angebrachter Ultraschalltransmitter soll die Positionierung erfolgen. Da sich das Fahrzeug in der Nähe des Bodens oder gar direkt auf dem Boden bewegt, ist das zugrundeliegende Szenario mit den vorigen vergleichbar. Zunächst wird aber der allgemeine Fall betrachtet, in dem sich das Vehikel über dem Grund aufhält. Das Fahrzeug ist mit einem Ultraschalltransmitter ausgerüstet, der in regelmäßigen Zeitabständen Signale sendet. Sobald ein am Meeresgrund angebrachter Knoten ein derartiges Signal empfängt, wartet er ein vordefiniertes Zeitintervall δ_r und antwortet dann mit einem eigenen Ultraschallsignal. Nun lässt sich der Positionsraum des Tranceivers mit $c\Delta_t = |T-R| + |v\Delta_t + T-R|$ auf einen Kreis einschränken. Dabei sind T und R die Positionen des Tranceivers am Boden bzw. am Fahrzeug. Mit einer zweiten und dritten Messung kann der tatsächliche Ort des Empfängers bestimmt werden. Da in dem vorliegenden Versuch jedoch von einer im Verhältnis zur Schallgeschwindigkeit im Wasser kleinen Vehikelgeschwindigkeit ausgegangen wird, fällt der Term $v\Delta_t$ komplett weg. Um das Problem mathematisch zu erfassen, wird eine Beobachtungsfunktion h eingeführt, die abhängig vom Zustand X ist. Damit der Zustand aufgrund der Beobachtungsfunktion geschätzt werden kann, wird h in Abhängigkeit von X folgendermaßen beschrieben:

$$z_i(k) = h_{i,k}(X)$$

mit $z_i(k)$ times of flight Beobachtung von der Position k bezüglich des Transponders i .

Dann werden alle derartigen Beobachtung in einer Matrix Z_t kokateniert.

Der Vektor X enthält dabei die geschätzten Positionen des Vehikels aller Durchgänge und die vermuteten Orte der Transponder. Zunächst wird eine Matrix $Z_t = h(X)$ aus allen beobachteten Antworten $z_i(k)$ gewonnen. Um daraus eine optimale Schätzung zu erhalten, wird ein nichtlinearer Kleinste-Quadrate-Ansatz verwendet. Dieser löst wiederholt folgendes aus h linearisierte System von Gauss-Newtongleichungen:

$$H\delta x = -J^T W \delta z,$$

bei dem H und J die Hesse- bzw. Jakobimatrix darstellen,

$W = \begin{bmatrix} R^{-1} & 0 \\ 0 & Q^{-1} \end{bmatrix}$ die Beobachtungsgewichtungsmatrix und δz den Beobachtungsfehler.

Im Versuch werden zusätzlich unbrauchbare und zu leise Signale ausgefiltert. Es wird ein Kalmanfilter eingesetzt, der ganze zehn Zustände beinhaltet. Die zu absolvierende Teststrecke hat eine Länge von 50 Metern und besteht aus 150 individuellen Positionen. Vier Transponder werden an verschiedenen Stelle angebracht. Mittels DGPS kann die gemessene mit der tatsächlichen verglichen werden.

Als Resultat konnten die vier Transponder ausreichend genau lokalisiert werden. Dabei gab es einen maximalen Fehler von 2.3 % und einen minimalen von 1.3 %. Die Methode scheint zwar für ein derartiges Szenario gut gerüstet, bei schnelleren Bewegungen muss jedoch die Geschwindigkeit des Fahrzeugs stärker berücksichtigt werden. Außerdem sind für die komplexen mathematischen Berechnungen leistungsstarke Rechner vonnöten. Dies kann nicht immer gewährleistet werden.

7 Fazit

Wie deutlich zu erkennen ist, gibt es im Forschungsbereich SLAM zahlreiche Ansätze, die teilweise in völlig unterschiedliche Richtungen gehen.

Die ersten praktischen Versuche gehen immer von bereits vorliegenden, sich von der Umgebung abhebenden Merkmalen aus. Hindernisse und Wände stellen besondere Landmarken dar, die in eine Karte eingetragen werden. Aus den Kameradaten werden sodann die Winkel extrahiert, die zu den Landmarken gegeben sind. Da die Kameras und die damit verbundene Bildverarbeitung jedoch sehr ungenaue Ergebnisse liefern, werden Filter eingesetzt, die die Schätzungen verbessern.

Nichtsdestotrotz gelten die Versuche als eher ernüchternd. Da Winkelmessungen zu ungenau sind und mittlerweile alternative Sensorsysteme existieren, setzt man den Schwerpunkt der Forschung in Richtung Entfernungsmessung. Die neue Philosophie wird durch verschiedene, neuartige Ansätze verfolgt. Durch Infrarot wird nicht nur die diskrete, sondern auch die absolute Lokalisierung ermöglicht. In der Arbeit [1] wird eine absolute Lokalisierung mithilfe von Intensitätsmessungen beschrieben.

Schließlich setzt sich durch Ultraschall und zugrundeliegende times-of-flight-Methoden das Multilaterationsverfahren durch. Positionsbestimmungen über Entfernungsmessungen werden auch durch Laserrangefinder ermöglicht. In welchem Szenario welches Verfahren besser geeignet ist, muss individuell entschieden werden. Ein Beispiel stellt das Ultraschallverfahren [9] in Meeren dar. Dieses erscheint zwar zunächst als eine annehmbare Lösung, wirft jedoch weitere Fragen auf. Gerade die mit Ultraschall kommunizierenden Wale und Delfine könnten durch Ultraschallsignale irritiert und damit in ihrem Lebensraum gestört wer-

den.

Da SLAM bis jetzt noch nicht ausgereift ist, wird sich erst in den kommenden Jahren herausstellen, ob sich intelligente Sensornetzwerke wie die aus dem Relateprojekt [8] von den anderen SLAM-Praktiken abheben können.

Literaturverzeichnis

- [1] Mike Hazas Albert Krohn, Michael Beigl and Albrecht Schmidt. Using fine-grained infrared positioning to support the surface-based activities of mobile users.
- [2] Alan Jones Andy Ward and Andy Hopper. A new location technique for the active office, 1997.
- [3] Daniel Boley and Karen Sutherland. A rapidly converging recursive method for mobile robot localization.
- [4] Kavitha Muthukrishnan Carl Fischer, Mike Hazas and Hans Gellersen. Ultrasound-aided pedestrian dead reckoning for indoor navigation. 2008.
- [5] Frank Dellaert Dieter Fox, Wolfram Burgard and Sebastian Thrun. Monte carlo localization: Efficient position estimation for mobile robots.
- [6] A. Eliazar and R. Parr. Dp-slam: Fast, robust simultaneous localization and mapping without predetermined landmarks, 2003.
- [7] George Kantor Joseph Djugash, Sanjiv Singh and Wei Zangh. Range-only slam for robots operating cooperatively with sensor networks. *International Conference on Robots and Automation*, 2006.
- [8] Hans Gellersen Mike Hazas, Christian Kray and Gerd Kortuem. A relative positioning system for co-located mobile devices.
- [9] P. Newman and J. Leonard. Pure range-only sub-sea slam.
- [10] John Spletzer. A new approach to range-only slam for wireless sensor networks.
- [11] Phillip Tomé Valérien Renaudin, Okan Yalak and Bertrand Merminod. Indoor navigation of emergency agents, 2007.
- [12] Roy Want, Veronica Falcao, and Jon Gibbons. The active badge location system. *ACM Transactions on Information Systems*, 10:91–102, 1992.
- [13] Wikipedia-Community. Global positioning system. *Wikipedia*, 2008.
- [14] Wikipedia-Community. Taylorreihe. *Wikipedia*, 2008.
- [15] Wikipedia-Community(en). Inertial navigation system, 2008.
- [16] Wikipedia-Community(en). Multilateration. *Wikipedia*, 2008.
- [17] Wikipedia-Community(en). Radio frequency identification. *Wikipedia*, 2008.

“Unterstützende“ Technologien für den Einsatz in Gefahrensituationen - Sensorik und Kommunikation

Wolfgang Heetfeld

Universität Karlsruhe (TH) upbdx@stud.uni-karlsruhe.de

1 Einleitung

Rettungskräfte sehen sich täglich mit einer Vielzahl an Gefahrensituationen konfrontiert. Sie werden gerufen, wenn Menschenleben bedroht sind und Eigentum geschützt werden muss. Besondere Anforderungen stellen dabei Katastrophenszenarien dar. In der Ausarbeitung wird gezeigt wie es möglich ist, durch den Einsatz von Sensorik und Kommunikationseinheiten, gezielt Hilfe zu leisten. Die wichtigsten Problemstellungen sind dabei:

- Aufklären der Lage
- Überwachen von Einsatzkräften
- sowie Koordinieren des Einsatzes

Im Folgenden werden speziell Katastrophen betrachtet. Allein im Jahr 2007 wurden 414 Naturkatastrophen gezählt, die einen Schaden von knapp 58,4 Millionen Euro verursachten. Dabei waren mehr als 211 Millionen Menschen von den Gefahren unmittelbar betroffen, 16847 Personen kostete es das Leben. Diese Statistik [Scheuren 07] bestätigt den globalen Aufwärtstrend der weiter wachsenden Zahl an Naturkatastrophen, die Jahr für Jahr auftreten. Einer der Gründe dafür ist sicherlich auch der Klimawandel. Obwohl Europa ein Kontinent ist, der von Naturkatastrophen eher selten getroffen wird, ist der Schaden enorm. Zwar kamen 2007 lediglich 1% aller Opfer aus Europa, der Anteil am wirtschaftlichen Gesamtschaden betrug allerdings stolze 27% und das bei nur 65 gezählten Katastrophen. Mit Ausnahme von Waldbränden gehören Feuerszenarien nicht zu den aufgeführten Naturkatastrophen. Aus der World Fire Statistic für das Jahr 2004 [Brushlinsky 06] geht hervor, dass allein bei Bränden, in den weltweit 33 beobachteten Ländern, 30170 Menschen starben.

Die besten Methoden, um auf derlei Gefahren vorbereitet zu sein, sind sicherlich präventiver Natur. Durch Frühwarnsysteme können potentielle Gefahren rechtzeitig erkannt, eventuelle Vorkehrungen getroffen sowie Menschen und Tiere in Sicherheit gebracht werden. Was aber, wenn eine Katastrophe bereits eingetroffen ist und die präventiven Vorkehrungen sogar versagt haben? Dieser Fragestellung wird im Weiteren auf den Grund gegangen werden. Speziell wird angeschaut, welche unterstützenden Technologien, gerade in Form von Sensoren und Kommunikationstechniken, in einem solchen Gefahrenumfeld eingesetzt werden und in wieweit diese gesetzten Anforderungen gerecht werden.

2 Grundlagen

2.1 Technologieeinsatz

Bevor über Technologien gesprochen werden kann, muss erst einmal Klarheit darüber herrschen was damit gemeint ist. Laut der freien Enzyklopädie Wikipedia [Wikipedia 08], wird der Begriff Technologie als Synonym für Technik verwendet und meint das Wissen um eingesetzte Technik. Technologien werden benutzt um bestimmte Aufgaben zu erleichtern und oder zu beschleunigen. Sie müssen aber auch auf Akzeptanz bei den Einsatzkräften stoßen. Es ist deshalb wichtig intuitive Schnittstellen zwischen Benutzer und Technologie zu schaffen. Wir wollen auf diese Fragestellung aber nicht weiter eingehen.

2.2 Sensoren

Sensoren sind in der Lage Umgebungszustände zu erfassen, die dann ausgewertet und dem Menschen zugänglich gemacht werden können. Nachfolgend wird kurz auf einige Sensoren eingegangen, die im Rahmen von Katastropheneinsätzen Anwendung finden:

Beschleunigungssensor: Ein Beschleunigungssensor misst die Beschleunigung und deren Richtung die ein Objekt erfährt. Dazu ermittelt er die auf einen Körper wirkende Trägheitskraft.

CCD-Bildsensor: Charge-Coupled Device (CCD)-Sensoren sind die weitverbreitetsten Sensoren zur Aufzeichnung von Bildern. Der Sensor besteht aus mehreren nebeneinander angeordneten CCD - Elementen. Fällt auf ein solches Element Licht, so werden Elektronen freigesetzt. Je mehr Licht einfällt, desto mehr Elektronen setzt der Sensor frei. Durch spezielle Filter trifft nur eine bestimmte Wellenlänge, entweder Rotes, Grünes oder Blaues Licht, auf das CCD - Element auf. Kombiniert man nun drei aufeinanderfolgende Elemente, so erhält man die Farb- und Helligkeitswerte eines Pixels. Das gewonnene Bild gleicht der Wahrnehmung der menschlichen Augen und ist empfindlich gegenüber Verschmutzung.

Tensiometer: Sensor, der die Feuchtigkeit an der Oberfläche eines Gegenstands misst. Ein Tensiometer besteht aus zwei Kontakten die in einer definierten Entfernung zueinander stehen. Da Wasser Strom leitet fließen, je nach Feuchtigkeit, mehr oder weniger viele Elektronen von einem zum anderen Kontakt. Daraus lässt sich die Feuchtigkeit auf dem gemessenen Objekt bestimmen. Unter anderem kann ein solcher Sensor zur Bestimmung der Leitfähigkeit der Haut benutzt werden. Die Hautleitfähigkeit gibt an, wie viel Stress ein Mensch gerade ausgesetzt ist.

Geruchssensor: Geruchssensoren messen das Vorkommen bestimmter Moleküle in der Luft. Die eingesetzte Technik ist dabei von Sensor zu Sensor verschieden. Prinzipiell gibt es aber bei den meisten Sensoren mehrere Felder die jeweils nur spezielle Geruchsmoleküle binden. Aus der Kombination und der Anzahl der gebundenen Moleküle entsteht dann eine Art Geruchsmuster das einen Geruch identifiziert.

GPS-Sensor: Sensor, der die von Satelliten ausgestrahlten Signale empfängt, aus deren Signallaufzeit die aktuelle Position auf der Erde berechnet wird. Zur Ermittlung wird das Signal von vier Satelliten benötigt, wodurch Genauigkeiten von ca. 15 Meter erreicht werden. Durch die Erweiterung Differential Global Positioning System (DGPS), bei der eine Referenzstation das Satellitensignal empfängt, kann die Genauigkeit der Ortung auf unter 5 Meter verbessert werden. Dies funktioniert, da die Position der Referenzstation bekannt ist und so der Fehler der aktuellen Positionsbestimmung ermittelt werden kann. Das Global Positioning System (GPS)-Signal lässt sich durch Wolken oder Niederschlag kaum beeinflussen. In Gebäuden ist ein Empfangen des Signals aber schwierig und führt zu großen Abweichungen in der Genauigkeit der Positionsbestimmung.

Infrarotsensor/-kamera: Ein Infrarotsensor misst die Strahlung im Bereich von 0,7 bis 18 μm . Dazu wird ein Infrarotstrahl ausgesandt. Da die Intensität der Reflektion abhängig von der Temperatur des Körpers ist, kann man daraus die Wärme an der Oberfläche berechnen. Infrarotsensoren können so die für das menschliche Auge unsichtbare Wärmestrahlung sichtbar machen. Sie finden deshalb unter anderem in Wärmebildkameras Anwendung. Aktuelle Sensoren sind in der Lage Temperaturen zwischen 0 und 1000 $^{\circ}\text{C}$ zu messen und überstehen kurzfristige Umgebungstemperaturen von bis zu 375 $^{\circ}\text{C}$. Aus den Messwerten ist es möglich ein Graustufenbild mit 256 Grautöne zu erzeugen. Da allerdings der Mensch selbst nicht in der Lage ist so viele Helligkeitsstufen zu differenzieren, wird das Bild meist in eine sogenannte Falschfarben-Darstellung umgewandelt. Hierbei werden die verschiedenen Helligkeitswerte durch Änderung der Farbe sichtbar gemacht. Infrarotsensoren sind empfindlich gegenüber Verschmutzungen und müssen auf das darzustellende Objekt ausgerichtet werden.

Mikrofon: Ein Mikrofon wandelt, durch die Luft übertragenen Schall, in ein elektrisches Signal um. Dabei wird eine dünne, elastische Membran, infolge des Schalldrucks, in Schwingung versetzt. Ihre, den Schwankungen des Schalldrucks folgenden Bewegungen werden erfasst und durch einen Wandler in einen Spannungsverlauf umwandelt. Es existiert eine Vielzahl von Mikrofonbauarten, die sich in der Größe, der Empfindlichkeit oder des zu nutzenden Frequenzbereichs unterscheiden.

Lambdasonde: Sensor, der den Sauerstoffgehalt in einem gemessenen Gas bestimmt. Zur Messung kann eine sogenannte Widerstandssonde benutzt werden.

Das Sensorelement besteht aus halbleitender Titandioxidkeramik. Umgebender Sauerstoff besetzt nun die Fehlstellen und reduziert damit die Anzahl freier Ladungsträger, sodass der Widerstand größer wird je mehr Sauerstoff sich in der Umgebung befindet.

3D - Laserscanner: Ein Laserscanner misst die Entfernung von Gegenständen. Dazu wird ein kurzer Laserimpuls ausgestrahlt und die Zeit berechnet bis Teile des Signal zurückgekommen sind. Daraus erhält man eine Art Punktwolke als dreidimensionales Abbild der gescannten Umgebung. Ein solcher Laserscanner ist in der Lage mehr als 100.000 Messpunkte pro Sekunde zu errechnen. Da er mit Lichtgeschwindigkeit arbeitet sind auch große Entfernungen abtastbar. Dabei erreicht er eine Genauigkeit von bis zu 1 mm. Durch die kurzen Pulse ist der Laser zudem weniger anfällig gegen Wettereinflüsse, wie Nebel oder Niederschlag. Probleme hat der Laserscanner allerdings bei Glas, durch das er einfach hindurch geht.

Pulsoxymeter: Das Pulsoxymeter ermittelt die Sättigung des Bluts mit Sauerstoff. Dazu benutzt es zwei Infrarotdioden welche eine Körperstelle bestrahlen. Da mit Sauerstoff gesättigtes Hämoglobin anders gefärbt ist als Ungesättigtes, wird das Rotlicht unterschiedlich absorbiert. Ein auf der gegenüberliegenden Seite angebrachter Fotosensor misst diesen Unterschied. Bei diesem Verfahren treten Messfehler auf, wenn das Pulsoxymeter an lackierten Fingernägeln angebracht ist. Auch eine Kohlenstoffmonoxidvergiftung lässt sich nicht erkennen. Da Kohlenstoffmonoxid eine stärkere Bindung zum Hämoglobin hat als Sauerstoff, wird dieser verdrängt. Das Pulsoxymeter kann nicht unterscheiden, ob das Hämoglobin nun mit Sauerstoff oder Kohlenstoffmonoxid gesättigt ist. Neben der Blutsättigung kann zudem auch der Puls des Menschen gemessen werden.

Temperatursensor: Sensor, der die Temperatur in eine elektrische Größe umwandelt. Beispielsweise benutzt man dazu Kaltleiter, die ihren Widerstand bei steigender Temperatur erhöhen. Kaltleiter aus Platin können so Temperaturen zwischen -200 und 800 °C messen.

Ultraschall-Transducer: Es wird ein kurzer, hochfrequenter Schallimpuls ausgesandt, der sich im Raum ausbreitet. Trifft die Welle auf ein Objekt, so wird sie reflektiert und kommt als Echo wieder beim Sensor an. Aus der Zeitdifferenz zwischen dem Aussenden des Signals und dem Empfangen des Echos lässt sich die Entfernung zum Objekt berechnen. Dabei sind Ultraschall-Transducer relativ unempfindlich gegenüber Verschmutzungen wie Staub, Rauch oder Dunst. Sie erreichen Messgenauigkeiten von 0,11 mm. Auch die Beschaffenheit oder die Farbe des zu erkennenden Objekts ist zu vernachlässigen.

Anemometer: Ein Anemometer misst die Geschwindigkeit und Richtung des Winds. Dazu können beispielsweise drei Ultraschall-Transducer benutzt werden, die an den Ecken eines gleichseitigen Dreiecks angebracht sind. Sie senden

und empfangen abwechselnd einen kurzen Schallimpuls von einem Ultraschall-Transducer zum Anderen. Aus der Abweichung der Laufzeit lässt sich daraus Windrichtung und Windgeschwindigkeit ermitteln. Diese Art der Windmessung ist frei von mechanischen Problemen, wie sie bei herkömmlichen Anemometern auftreten. Die Abweichung der Messergebnisse liegt im Bereich von $\pm 3\%$.

2.3 Kommunikationstechniken

Um Informationen zwischen zwei oder mehreren Personen und Geräten austauschen zu können gibt es eine Vielzahl an Kommunikationstechniken. Im Weiteren werden einige für die Szenarien denkbare Technologien kurz erklärt werden:

Bluetooth: Drahtlose Funktechnik zur Übertragung von Daten im 2,4 GHz Bereich. Die maximale Übertragungsrate liegt bei 2,1 Mbit/s. Handelsübliche Geräte haben eine Reichweite von ca. 10 Meter im Freien. Theoretisch sind bei Sichtkontakt aber auch Reichweiten um die 100 Meter möglich. Auch hier ist die Reichweite stark von der Beschaffenheit der Hindernisse abhängig. Um mit mehreren Teilnehmern zu kommunizieren, können in einem sogenannten Piconet bis zu 255 Geräte zusammengefasst werden, von denen allerdings immer nur 8 gleichzeitig aktiv sein können. Zum Sparen von Energie besitzt Bluetooth zudem mehrere Stromsparmodi in welche die Geräte versetzt werden, wenn gerade keine Daten zu übertragen sind.

GSM: Digitaler Mobilfunk zur Telefonie sowie zur Datenübertragung. In Europa benutzt Global System for Mobile Communications (GSM) das 900 MHz - Frequenzband und erreicht eine Datenübertragungsrate von 14,4 kbit/s. Um die Datenrate weiter zu erhöhen wurden Erweiterungen entwickelt. So kommt man beispielsweise mit dem Einsatz von General Packet Radio Service (GPRS) auf eine maximale Datenrate von 171,2 kbit/s. GSM schafft im Freien eine Reichweite von ca. 35 km pro Zelle. Wobei die Reichweite je nach Art und Anzahl an Hindernissen schwankt. Zudem unterstützt GSM Handover, sodass während eines Gesprächs oder einer Datenübertragung die Zelle gewechselt werden kann.

Satellitenkommunikation: Telekommunikation zwischen zwei möglicherweise mobilen Bodenstationen zur Datenübertragung. Man unterscheidet zwischen Low Earth Orbit (LEO)-, Geostationary Earth Orbit (GEO)- und Medium Earth Orbit (MEO)-Systemen, die sich in den Entfernungen ihrer Umlaufbahnen unterscheiden. Weiter entfernte Umlaufbahnen haben den Vorteil, dass ein riesiges Gebiet über einen einzigen Satelliten kommunizieren kann, wobei dadurch auch die Signallaufzeit größer wird. Satelliten auf näheren Umlaufbahnen sind dagegen für Positionen auf der Erde nur zwischen 10 und 40 Minuten sichtbar, wodurch viele Handover nötig sind.

UMTS: Weiterentwicklung des Mobilfunkstandards GSM zur Telefonie und Datenübertragung im Bereich von 1900 - 2100 MHz. Universal Mobile Telecommunications System (UMTS) erreicht eine Datenübertragungsrates von 7,2 Mbit/s. Mit der Erweiterung High Speed Downlink Packet Access (HSDPA) werden es sogar bis zu 14,4 Mbit/s. Die Zellgröße nimmt mit steigender Auslastung ab und kann bis ca. 10 km weit reichen. Wie GSM unterstützt auch UMTS Handover von einer zur anderen Zelle.

UWB: Drahtlose Nahbereichskommunikation zur digitalen Funkübertragung. Dazu nutzt Ultra Wideband Communication (UWB) den Frequenzbereich zwischen 3,1 und 10,6 GHz. UWB hat mit 10-50 Meter zwar nur eine geringe Reichweite, schafft in diesem Bereich aber Übertragungsrates zwischen 480 und 1320 Mbit/s.

WLAN: Wireless Local Area Network (WLAN) bezeichnet ein drahtloses Funknetz zur Übertragung von Daten. Dazu nutzt WLAN hauptsächlich das 2,4 GHz-Band. Man unterscheidet zwischen dem Infrastruktur- und dem Ad-hoc-Modus. Während bei der Infrastruktur-Methode die Kommunikation über eine Basisstation koordiniert wird, sind im Ad-hoc-Modus alle Stationen gleichwertig. WLAN schafft heute bereits Übertragungsrates von bis zu 589 Mbit/s. Handelsübliche Endgeräte haben eine Reichweite von 30-100 Meter im Freien. Mit speziellen Antennen lässt sich die Reichweite sogar auf 100-300 Meter auf freien Flächen bzw. bis maximal 90 Meter durch geschlossene Räume erhöhen, wobei die tatsächliche Reichweite stark von der Art der Hindernisse abhängig ist. Mit einer Basisstation können 253 weitere Geräte verbunden werden.

Zigbee: Funktechnik zur drahtlosen Übertragung von Daten im 2,4 GHz Bereich. Die Übertragungsrates von 250 Kbit/s ist zwar langsamer als Bluetooth, dafür aber auch im Verbrauch sparsamer. Zigbee schafft Übertragungsrates zwischen 10 und 100 Meter, ähnlich wie Bluetooth. Zur gemeinsamen Kommunikation können bis zu 240 Geräte in einem sogenannten Zigbee - Personal Network (PAN) verbunden werden, wobei es pro PAN immer genau einen Koordinator gibt der das Netz verwaltet.

3 Katastrophenszenarien

3.1 Großbrand

Es handelt sich im Folgenden um einen Großbrand, weil hier im Gegensatz zu einem kleineren Brandeinsatz meistens mehrere Feuerwehreinheiten gleichzeitig benötigt werden. Hier ist eine gute Absprache untereinander von Nöten, die den Einsatz von Kommunikationsmitteln unbedingt erfordert. Aus Gründen der besseren Lesbarkeit wird im weiteren Verlauf auf die gleichzeitige Verwendung männlicher und weiblicher Sprachformen verzichtet. Die verwendeten

männlichen Nomen schließen die weiblichen Formen grundsätzlich mit ein. Ist also von Feuerwehrmännern die Rede, sind natürlich sowohl Feuerwehrmänner als auch Feuerwehrfrauen gemeint.

Problematik: Wenn Einsatzkräfte zu einem Großbrand gerufen werden, liegt bereits ein Desaster vor. Die für die einzelnen Helfer zur Verfügung stehenden Informationen sind jedoch meist nur spärlich. Die Herausforderung speziell für die Einsatzleitung besteht nun darin, sich schnellstmöglich einen Überblick über die aktuelle Lage zu verschaffen und diese zu klären. Wobei ihr Hauptaugenmerk zuerst auf der Rettung von Menschenleben und dann auf der Beseitigung der Gefahrensituationen liegt.

Ausrüstung: Der wichtigste Ausrüstungsteil, der die Einsatzkräfte vor den Gefahren am Einsatzort schützt ist, die Persönliche Schutzausrüstung (PSA) [Wikipedia 08]. Eine PSA umfasst einen Feuerwehr-Schutzanzug, einen Feuerwehrhelm mit Nackenschutz, Feuerwehr-Schutzhandschuhe sowie Feuerwehrstiefel. Je nach Aufgabenbereich und Gefahr für einen Feuerwehrmann stehen weitere Ausrüstungsgegenstände zur Verfügung. Die Feuerwehrpersonen die bei einem Brand ins Gebäude vordringen nennt man im Feuerwehrumfeld Angriffstrupp [Wikipedia 08]. Sie besitzen als zusätzliche Ausrüstungsgegenstände etwa einen Pressluftatmer oder eine Axt. In voller Montur bringt es die Schutzkleidung eines Angriffstrupps mit der kompletten Ausrüstung auf über 40 kg Gewicht.

Gefahren: Gerade der Angriffstrupp ist bei einem Feuerwehreinsatz besonders gefährdet, da er direkt mit den Gefahren konfrontiert ist. Wir wollen, zum besseren Überblick, die möglichen Gefahren einmal betrachten:

- **Giftiger Rauch:** Die größte Gefährdung stellt der giftige Rauch dar. Die in ihm enthaltenen Atemgifte wie Kohlenstoffmonoxid, Cyangase und andere, führen schon nach wenigen Atemzügen zur Bewusstlosigkeit. Die Haupttodesursache von Brandopfern ist deshalb das Ersticken. Trotz des Pressluftatmers, der die Feuerwehrmännern während des Einsatzes mit Sauerstoff versorgt, stellt der entstehende Rauch eine enorme Behinderung für die Einsatzkräfte dar. Zwar sind sie vor den giftigen Gasen geschützt, allerdings nimmt ihnen der dichte Rauch die Sicht, so dass sie nur sehr wenig sehen können. Der Angriffstrupp tastet sich in einer solchen Situation beinahe blind durch die Räume.
- **Hitze:** Als weitere Gefährdung ist die enorme Hitze, die ein Brand erzeugt, zu nennen. In Folge der hohen Temperaturen kann es zu Austrocknungen oder Verkohlungen des Organismus kommen. Ab einer Temperatur von 45 °C spürt ein Mensch Schmerzen, bei 55 °C kommt es zu ersten Verbrennungserscheinungen. Moderne PSA können die Trägerperson kurzfristig vor Umgebungstemperaturen bis zu 800 °C schützen. Trotzdem kann eine Feuerwehrperson damit nicht einfach so durch ein Feuer spazieren, da hier Temperaturen von 1000 °C und mehr keine Seltenheit sind.

- **Backdrafts:** Problematisch sind auch so genannte "Backdrafts" [Hinck 07]. Durch plötzliches Hinzufügen von Sauerstoff zu einem Schwelbrand, wie es beispielsweise beim Öffnen einer Tür geschehen kann, kommt es hier zu einer Explosion.
- **Panikreaktionen:** Des Weiteren können auch Panikreaktionen auftreten, wenn eine Person ohne Vorwarnung, vom einen Moment auf den Anderen, die Kontrolle verliert [Wikipedia 08]. Sie treten hauptsächlich dann auf, wenn sich ein Mensch in einer gefährlichen Situation bedroht fühlt. Auch ausgebildete und speziell geschulte Einsatzkräfte sind im Ernstfall vor solchen Angstreaktionen nicht gefeit. Durch gezielte Schulungen kann die Gefährdung aber verringert werden. Grundsätzlich ist das Phänomen aber eher selten und wird deshalb nicht näher erörtert.
- **Ausgehen der Atemluft:** Durch das Gewicht des Atemschutzgeräts und anderer schwerer Ausrüstungsgegenstände sowie der körperlichen Arbeit steigt der Sauerstoffverbrauch eines Feuerwehrmanns. In der Praxis hat sich gezeigt, dass eine Einsatzkraft mit einem Pressluftatmer eine Verweildauer von ca. 20 min. am Einsatzort hat. Durch Risse im Atemschlauch kann diese Zeit aber dramatisch sinken. Sind die Sauerstoffreserven in der Gasflasche aufgebraucht, so ist die Feuerwehrperson den giftigen Dämpfen schutzlos ausgeliefert. Die Feuerwehr in Deutschland schreibt es deshalb vor, dass bei jedem Einsatz eine Person zum Dokumentieren der verbleibenden Sauerstoffversorgung abgestellt wird [Hinck 06].
- **Verirren im Gebäude:** Die in Folge des Rauches oft sehr schlechte Sicht führt ohne Hilfsmittel dazu, dass sich auch eine geschulte Einsatzkraft in der fremden Umgebung verirren kann. Um zusammenbleiben zu können und später den Weg aus dem Gebäude wieder zu finden, sind deshalb alle Personen durch eine so genannte Life-Line verbunden [Klann 07]. Die Life-Line ist ein Seil, das die Rettungskräfte mit ins Gebäude nehmen. Der Feuerwehrmann hält sich so immer einen Fluchtweg offen, der den direkten Kontakt mit den Brandherden vermeidet. Jedoch kommt es vor, dass die Life-Line an Hindernissen hängen bleibt oder in Kontakt mit dem Feuer teilweise zerstört wird.

Denkbarer Einsatz von Sensoren und Kommunikationstechniken zur Gefahrenminimierung: Bisher wurde erläutert, welche Herausforderungen ein Feuerwehreinsatz an die Einsatzkräfte stellt. Speziell ist auf Gefährdungen für den Angriffstrupp eingegangen worden, da dieser unmittelbar mit den meisten Gefahren konfrontiert ist. Nun wird betrachtet welche Sensoren und Kommunikationstechnologien für die jeweiligen Anforderungen in diesem Szenario denkbar sind, um den Feuerwehrmann zu unterstützen und die Gefährdung im Einsatz zu verringern.

- **Aufklärung der Lage:** Um ein effektives Vorgehen gewährleisten zu können ist umfangreiches Wissen über die aktuelle Situation von Nöten. Beispielsweise ist es wichtig die genaue Anzahl, an vom Feuer eingeschlossener Menschen, zu kennen und es wäre hilfreich einen Lageplan des betroffenen Gebiets zu

erhalten. Falls es darüber keine Angaben gibt, müssen diese schnellstmöglich erbracht werden. Das Ausstatten der Feuerwehrlaute mit CCD-Bildsensoren könnte der Einsatzleitung helfen einen Überblick über die Lage zu bekommen. Fraglich ist jedoch, ob ein Kamerabild in dichtem Rauch überhaupt verwertbare Informationen liefert.

Ein anderer Ansatz wäre die Nutzung autonomer oder ferngesteuerter Roboter. Diese Roboter könnten gezielt in gefährdete Bereiche vordringen und dort die Lage aufklären. Wichtig hierfür wäre ein schnelles Vorrücken der mechanischen Helfer. Eventuell könnte man hierzu auch fliegende Aufklärer benutzen. Mittels Ultraschall-Transducer wäre es möglich auch in Räumen mit dichtem Rauch sicher zu navigieren. Für einen solchen Einsatz müssten die Bauteile des Roboters feuerfest sein und enorme Temperaturen aushalten können. Gerade bei der Elektronik, die im Allgemeinen sehr hitzeanfällig ist, darf diese Problemstellung nicht unterschätzt werden.

- **Überwachung von Vitalfunktionen:** Zudem kommt es vor, dass eine Feuerwehrperson selbst Hilfe benötigt. Um in solchen Fällen schnell reagieren zu können, ist es wichtig zeitnah über die Gefährdung informiert zu werden. Deshalb könnten zur Überwachung des Feuerwehrmanns Pulsoxymeter eingesetzt werden, die den Herzschlag und die Sauerstoffsättigung im Blut ermitteln. Gelangen die Werte in einen kritischen Bereich kann die Einsatzleitung alarmiert werden, die dann Maßnahmen ergreifen kann.
- **Hitze:** Damit die Feuerwehrperson auf zu hohe Temperaturen aufmerksam gemacht wird könnten Temperatursensoren eingesetzt werden. In Abhängigkeit der Temperatur könnten dann akustische und visuelle Warnsignale gegeben werden.
- **Backdrafts und Rauch:** Ein Infrarotsensor, eingebaut in eine Wärmebildkamera, ließe versteckte Gefahren, wie etwa hinter Türen lauernde Backdrafts, frühzeitig erkennen. Auch wäre es möglich kältere Objekte, wie beispielsweise einen von Flammen eingeschlossenen Menschen, inmitten eines Feuers zu erkennen. Selbst in dichtem Rauch ist es mit Hilfe einer Wärmebildkamera möglich die Umgebung wahrzunehmen. Allerdings darf ein solcher Sensor nicht zu schwer ausfallen und sollte nach Möglichkeit an der Feuerwehrperson befestigt werden, damit diese die Hände für den Einsatz frei hat.
- **Panikreaktionen:** Über die Messung des Hautwiderstands wäre man in der Lage den Stressfaktor einer Person zu überwachen. Dazu könnte man einen Tensiometer verwenden. Wird der Wert zu hoch, könnte die Einsatzleitung die betroffene Einsatzkraft zurückordern um eine mögliche Panikattacke zu vermeiden.
- **Verirren im Gebäude:** Die ständig mitgeführte und den Feuerwehrmann behindernde Life-Line könnte durch GPS - Tracking ersetzt werden. Dabei wird periodisch die Position des Feuerwehrmanns durch einen GPS-Sensor ermittelt und aus den Messpunkten eine virtuelle Life-Line generiert. Diese könnte die Einsatzkraft bei Bedarf wieder aus dem Gebäude herausführen, ohne sich zu verirren.

- **Kommunikation:** Um die Geschwindigkeit bei der Aufklärung durch Roboter weiter zu erhöhen, wäre statt eines einzelnen Roboters auch ein ganzer Roboterschwarm denkbar. Die Agenten könnten dann untereinander kommunizieren, Ergebnisse austauschen und durch gezielte Aufgabenverteilung ihr Explorationstempo erhöhen. Auch für den Feuerwehrmann bräuchte die Datenübermittlung von gemessenen Sensordaten einen Mehrwert. Denkbar wäre hier der Einsatz bereits vorhandener Techniken wie GSM bzw. UMTS, da zumindest das GSM-Netz bereits flächendeckend ausgebaut ist. Sind GSM- bzw. UMTS-Netze am Einsatzort nicht verfügbar, wäre ein WLAN bzw. ein Wireless Metropolitan Area Networks (WMAN) eine Alternative. Ein solches Netz könnte rasch aufgebaut werden, indem jede Einsatzkraft mit einem WLAN-Gerät ausgestattet würde.

Tatsächlicher Einsatz von Sensoren und Kommunikationstechniken zur Gefahrenminimierung: Viele Technologien befinden sich derzeit noch in der Entwicklung. Natürlich kann an dieser Stelle keine allumfassende Betrachtung über den aktuellen Forschungsstand geliefert werden. Es werden vielmehr einige ausgewählte Sensor- und Kommunikationstechnologien vorgestellt, die mögliche Lösungsansätze zu obigen Problemstellungen liefern:

- **Aufklärung der Lage:** Erste Roboter sind bereits im Einsatz. Sie sind fähig in unwegsames Gelände vorzudringen und dort die Lage aufzuklären. Einige Modelle wie beispielsweise der TALON der amerikanischen Firma Foster-Miller [Bixby 07] (siehe Abb. 1) verfügen zudem über einen Greifer um beispielsweise Türen zu öffnen oder Gegenstände wegzuräumen. Zur Anwendung kommen hier vor allem Video- und Wärmebildkameras die CCD- bzw. Infrarotsensoren benutzen. Größtenteils, wie auch beim TALON der Fall, werden die Roboter von einem Operator gesteuert. Daneben existieren einige autonome Systeme, die aber den durch Operatoren unterstützten Robotern derzeit nicht das Wasser reichen können. Problematisch ist auch die Anfälligkeit dieser Systeme bei enormer Hitze. Hier ist ein Kompromiss zwischen Geschwindigkeit, Größe und Feuerfestigkeit des Roboters zu finden.



Abbildung 1. Roboter TALON der Firma Foster-Miller. Er kann per „plug and play“ mit bis zu sieben Sensoren bestückt werden.



Abbildung 2. LED - Warnanzeige des intelligenten Schutzanzugs der Firma VIKING. Links die Anzeige für den Hintermann und rechts die Anzeige für den Träger.

- **Überwachung von Vitalfunktionen:** Derzeit besitzt bereits jedes neu ausgelieferte Atemgerät einen integrierten Bewegungslosmelder auch Totmannmelder genannt. Dieser besteht aus einem Beschleunigungssensor der kontinuierlich die Beschleunigung misst. Bewegt sich der Träger des Atemschutzgeräts 25 sec. lang nicht ertönt ein Voralarm, der dann innerhalb von weiteren 15 sec. abgebrochen werden kann. Nach Ablauf der Voralarmzeit wird ein Alarm gesendet. Dieser kann ebenfalls per Knopfdruck ausgelöst werden.
- **Hitze:** Auch im Bereich der direkt an der Feuerwehrperson zum Einsatz kommenden Hilfsmittel gab es in den letzten Jahren etliche Fortschritte. Die dänische Firma VIKING hat beispielsweise einen Feuerwehr-Schutzanzug mit eingebauten Temperatursensoren auf den Markt gebracht [Eric 08]. Diese warnen über zwei Light Emitting Diode (LED)-Anzeigen (siehe Abb. 2), eine am Arm für den Träger und eine an der Schulter für den Hintermann, vor zu hohen Temperaturen. Die Sensoren messen sowohl die Umgebungstemperatur als auch die Temperatur im Inneren der Schutzkleidung. Erreichen die gemessenen Temperaturen die kritische Grenzen von 250 °C Umgebungsbzw. 50 °C Innentemperatur, beginnen die LED's langsam zu blinken. Für die betroffene Feuerwehrperson bedeutet dies, dass erste Schmerzen auftreten können. Ignoriert der Feuerwehrmann die Warnung, so beginnen die LED's ab einer Temperatur von 350 °C Außen- bzw. 67 °C Innentemperatur heftig zu blinken, wodurch eine für den Retter kritische Situation angezeigt wird, die zu Verbrennungen der Haut führen kann. Laut Hersteller übersteht die Schutzkleidung 25 Waschgänge unbeschadet. Problematisch könnte bei dieser Jacke die Visuallisierung sein. Im dichten Rauch eines Brands ist es möglich, dass eine solche Warnung nicht mehr zu sehen ist. Als Verbesserung könnte neben dem visuellen ein akustisches Signal benutzt werden. Auch die Waschbarkeit von 25 Waschgängen überzeugt noch nicht.
- **Backdrafts und Rauch:** Beim Schutzanzug der European Aeronautic Defence and Space Company (EADS) - Werksfeuerwehr (siehe Abb. 3) ist man noch weiter gegangen [Schulenburg 07]. Sie verfügt bereits über einen im Feuerwehrhelm integrierten Infrarotsensor als Wärmebildkamera, deren Bilder über ein Head-Up-Display angezeigt werden. Versteckt lauernde Ge-



Abbildung 3. Helm montierte Wärmebildkamera der EADS - Werksfeuerwehr.

fahren, wie etwa Backdrafts, können so rechtzeitig erkannt werden. Zudem ermöglicht es ein solcher Sensor auch in stark verrauchten Gebäuden etwas zu sehen. Die Kamera hat ein Gewicht von ca. 1,45 kg und schafft eine Einsatzzeit im Dauerbetrieb von 180 min. Die Messdistanz ist ausgelegt auf bis zu 15 m. Im Vergleich zu der Einsatzzeit einer Feuerwehroperson unter Atemschutz ist dies sehr lange. Die Firma Bullard, die selbst Feuerwehrausrüstungsgegenstände herstellt meint, dass die Zeit für Helmkameras noch nicht reif ist. So ist auf ihrer Homepage zu lesen, dass die derzeitigen Modelle noch zu schwer und unflexibel sind. So muss sich ein Feuerwehrmann etwa auf den Boden legen um unter ein Bett zu sehen. Durch den derzeit noch sehr hohen Preis¹ kann nicht jede Feuerwehroperson damit ausgestattet werden. Ein weitergeben ist bei einer Helmmontur aber unmöglich.

- **Verirren im Gebäude:** Als Alternative für die Life-Line wurde ein digitaler Fluchtweg mittels GPS - Tracking in Betracht gezogen. Leider erwies es sich in der Praxis, dass eine Ortung durch GPS in Gebäuden oft nicht zu gebrauchen ist. Eine interessante Alternative stellt deshalb das sogenannte LifeNet Konzept (siehe Abb. 4) dar [Klann 07]. Durch automatisches Auslegen von Sensorknoten entlang des Weges kann zu jedem Zeitpunkt ein Rückweg berechnet und den Einsatzkräften angezeigt werden, die an ihren Schuhen ebenfalls solche Knoten tragen. Jeder Sensorknoten ist eigenständig und verfügt über einen Prozessor, eine Kommunikationseinheit, einen Ultraschall-Transducer, einen Wärmesensor sowie weitere Schnittstellen. Beginnend bei einem dedizierten Sensorknoten am Eingang, wird so ein Netz aus Fluchtwegen aufgespannt das periodisch aktualisiert wird. Fällt ein Knoten aus, so werden die davon betroffenen Knoten darüber informiert, dass dieser Weg nicht mehr verfügbar ist. Ein Prototyp hierzu zeigt der Feuerwehroperson die kürzeste Route durch Pfeile an, welche über ein in die Atemschutzmaske integriertes Display dargestellt werden. Alternativ ist es auch möglich sich durch eine Sprachausgabe über ein Headset, dem Pfad entlang, leiten zu lassen. Dieses Konzept überzeugt weitgehend. Lediglich die Tatsache, dass die Sensorknoten bisher noch von Hand ausgelegt werden müssen ist zu verbessern.

¹noch kosten solche Kameras etwa 3000 Euro.

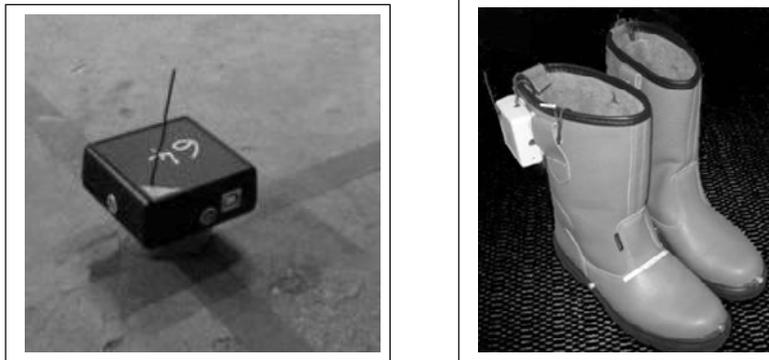


Abbildung 4. Mit Ultraschall bestückte Sensorknoten des LifeNet Projekts. Links ein Standardknoten, rechts ein am Stiefel integrierter Knoten. In ihrer Implementierung erreichen die Knoten eine Messgenauigkeit von 10cm bei einer Reichweite von 2,5m.

- **Kommunikation:** Zur einfachen Kommunikation benutzt die Feuerwehr in Deutschland derzeit häufig noch analogen Funk. Vereinzelt ist man² bereits auf den digitalen Nachfolger umgestiegen. Jedoch gibt es für die Übertragung von Bildern und anderen Sensordaten an die Leitzentrale in der Feuerwehrpraxis noch keine Nutzung. Ein in Frage kommendes Projekt wäre hierzu das Digital Ubiquitous Mobile Broadband Optimized Link State Routing (OLSR) Net (DUMBONET) [Kanchanasut 07]. Es setzt im konkreten Disasterfall hauptsächlich auf WLAN - Kommunikation, die durch einen spontanen Verbund unterschiedlicher Geräte im Ad-hoc Modus realisiert wird. Da es für den Einsatz in einem Katastrophenfall entwickelt wurde, ist es möglich mehrere solcher Ad-hoc-Netze über das Internet zu virtuellen Netzen zu verbinden. Dies wird durch die Verwendung eines Virtual Private Network (VPN) umgesetzt. Dazu ist vorgesehen, dass alle Teilnetze via Satellit miteinander verbunden werden. Eine solche Kommunikationsumgebung ließe sich auch dann aufbauen, wenn sämtliche fest installierte Technologien aufgrund der Katastrophe ausgefallen wären. Ein weiterer Vorteil dieser Herangehensweise ist, dass die Einsatzleitung nicht direkt vor Ort sein muss und trotzdem den Überblick über die Einsatzkräfte behalten kann. Einige mit DUMBONET kompatible Sensoren können bereits in Echtzeit Messdaten mitteilen und vergleichen. Auf Fragen der Sicherheit, wie etwa Verschlüsselung, Authentifizierung und Verifizierung der Daten wurde noch nicht eingegangen, was in der weiteren Entwicklung aber relevant sein wird. Ebenfalls problematisch ist die hohe Latenz, die aufgrund des Satelliteneinsatzes bei den Verbindungen auftritt. Daraus resultierend war die Sprachqualität im Test eher schlecht. Die Verwendung besserer Codecs und intelligenterer Fehlerkorrekturalgorithmen könnte dem entgegenwirken. Die lange Einrichtungszeit von knapp 3h für den Aufbau der Satellitenverbindung schließt den Einsatz von DUMBONET³ bei Großbränden aus. Die

²wie etwa bei der Werksfeuerwehr des Vorreiters EADS.

³zumindest in der vorgestellten Variante.

prinzipielle Verwendung eines Ad-hoc-Netzes für Feuerwehrleute scheint aber möglich. Während einige Probleme wie etwa die hohe Latenzzeit damit vom Tisch wären, ist als Hauptnachteil anzusehen, dass die Einsatzleitung nun tatsächlich vor Ort sein muss. Da dies beim „normalen“ Feuerwehreinsatz aber üblich ist dürfte diese Einschränkung zu verkraften sein.

3.2 Lawinenunglück

Neben dem Großbrand, ist auch das Szenario eines Lawinenunglücks zu betrachten. Wenn im Folgenden von Lawinen gesprochen wird, ist immer die Schneelawine gemeint.

Problematic: Auch hier ist eines der Hauptprobleme, schnell einen Überblick über die Katastrophe zu erlangen. Während ein Brand häufig in stark frequentierten Gegenden ausbricht und diverse Frühwarnsysteme dazu beitragen ihn schnell zu melden, ist dies bei Lawinen anders. Lawinenabgänge machen auch vor entlegenen Bergregionen nicht halt. Zwar ist es mittlerweile technisch durchaus möglich Lawinenabgänge zu lokalisieren, allerdings kann man nicht direkt herausfinden ob Menschen betroffen sind, die eventuell Hilfe benötigen. Ein Notruf von Augenzeugen oder selbst von der Lawine Betroffenen, ist in vielen Fällen die einzige Möglichkeit Rettungskräfte zu alarmieren [Wikipedia 08]. Wurde der erste Schritt getan, sodass Einsatzkräfte über den Abgang einer Lawine und von ihr betroffene Personen informiert sind, so ist das kritischste Moment der Zeitfaktor. Statistiken [Michahelles 03] zeigen, dass 92 % der Lawinenopfer die ersten 15 Minuten nach ihrer Verschüttung überleben. Nach 35 Minuten sinkt die Zahl schon auf 30 % und nach 130 Minuten leben nur noch 3 % der Opfer. Es ist deshalb wichtig, dass trotz der teilweise schwer zugänglichen Lage des Unglücks die Rettung schnell erfolgt. In der Praxis braucht die Bergrettung, aufgrund der Zeit die bis zur Alarmierung und Ausrückung verstreicht, meist mehr als 15 Minuten. Zufällig am Unglücksort befindliche und deshalb oft ungeschulte Personen sind häufig die ersten Helfer. Behindernd kommt hinzu, dass der Suchbereich in dem sich die Verschütteten befinden in vielen Fällen recht groß ist.

Ausrüstung: Derzeit sind zur Lokalisierung von Lawinenopfern sogenannte LVS-Geräte (siehe Abb. 5) im Einsatz. Die Geräte können jeweils als Sender oder Empfänger agieren. Ist bei einem Gerät der Empfängermodus aktiviert, so kann es Verschüttete lokalisieren und dadurch die Helfer zu den Opfern navigieren. Dies funktioniert aber nur, wenn der Verschüttete ebenfalls ein solches Gerät bei sich trägt. Ist dies der Fall dauert das Finden eines Lawinenopfers nicht länger als drei bis fünf Minuten [Michahelles 03]. Neben Opfern, die ein LVS-Gerät bei sich tragen, gibt es aber immer noch solche die so etwas nicht haben. Um auch diese Menschen schnell finden zu können setzt die Bergwacht speziell ausgebildete Hunde ein. Die Ausbildung eines solchen Lawinensuchhunds dauert allerdings mehrere Jahre und ist nicht ganz billig. Zum Ausgraben benutzt man eine herkömmliche Schaufel. Obwohl die Bergung einer verschütteten Person

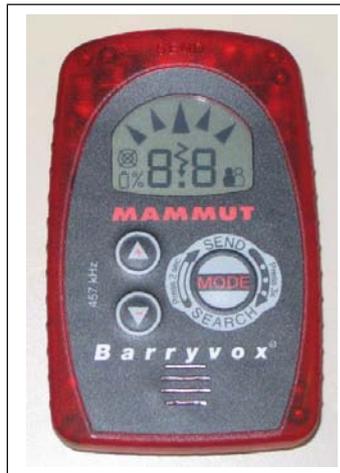


Abbildung 5. Herkömmliches LVS-Gerät wie es derzeit von Ski- und Snowboardfahren eingesetzt wird, die außerhalb präparierter Pisten unterwegs sind.

schnell vonstatten gehen sollte, muss beim Ausgraben mit äußerster Vorsicht vorgegangen werden, da man nicht weiß in welcher Position das Opfer liegt. Lawinenopfer können beim Ausgraben durch Schaufeln verletzt und letzte, die Opfer vor dem Erstickten bewahrende, Lufteinschlüsse zerstört werden.

Gefahren: Im Gegensatz zu den Gefahren denen sich die Feuerwehrleute⁴ bei einem Großbrand ausgesetzt sehen, ist das Gefährdungsrisiko für Bergretter eher gering. Unser Augenmerk wird deshalb auf die Bedrohungen für die Lawinenopfer gerichtet sein:

- **Ausgehen der Atemluft:** Die häufigste Todesursache bei einem Lawinenabgang ist das Erstickten. Dreiviertel der Todesopfer sterben, weil sie keine Luft mehr bekommen. Die komprimierte Schneedecke ist extrem hart, so dass eine verschüttete Person im schlimmsten Fall keine Möglichkeit hat sich zu bewegen. Ohne Luftblase besteht dann nur eine geringe Überlebenschance.
- **Kälte:** Weitere Gefahren lauern in der Kälte. Durch den Lawinenabgang können schwach befestigte Kleidungsstücke, wie Handschuhe, Mütze, etc. weggerissen werden. Aber auch, wenn dies nicht passiert kann es zu Unterkühlungen kommen die den Körper zusätzlich schwächen. An dieser Stelle sei noch der Bergungstod zu nennen. Wird eine unterkühlte Person bei der Rettung zu stark bewegt, kann dies den Kreislauf anregen, wodurch unterkühltes und sehr sauerstoffarmes Blut in die Organe fließt.
- **Verletzungen während des Lawinenabgangs:** Zudem ist es möglich, dass ein Mensch schon während des Lawinenabgangs verletzt oder getötet wird. Durch die hohen Geschwindigkeiten mit der eine Lawine ins Tal rast⁵,

⁴hier speziell wieder der Stoßtrupp.

⁵Lawinen erreichen Geschwindigkeiten von bis zu 450 km/h.

kann eine Kollision mit Bäumen, Felsen oder gar dem eigenen Skistock fatale Folgen haben. Glücklicherweise sind daraus resultierende schwere Verletzungen aber eher selten der Fall.

Denkbarer Einsatz von Sensoren und Kommunikationstechniken zur Lawinenrettung: Wir wollen nun betrachten welche Sensoren und Kommunikationstechnologien für die jeweiligen Anforderungen in diesem Szenario denkbar sind, um die Helfer bei der Rettung der Lawinenopfer zu unterstützen:

- **Absetzen eines Notrufs:** Eine mögliche Verschüttung durch eine Lawine könnte schon während deren Abgangs, durch eine Kombination aus Beschleunigungssensoren und Mikrofon, selbständig erfasst und dadurch die Bergrettung verständigt werden. Dabei würde ein Lawinenabgang erst als solcher detektiert werden, wenn ein lang anhaltender Lärm in Verbindung mit einer starken Beschleunigung erkannt wird. Obwohl die Detektion wann eine Person von einer Lawine erfasst wird wohl nicht ganz einfach ist, muss sie korrekt funktionieren. Neben hohen Kosten könnte ein Fehlalarm auch dazu führen, dass anderswo gebrauchte Rettungskräfte nicht zur Verfügung stehen. Zudem stellt sich die Frage mit welcher Kommunikationstechnik ein Notruf abgesetzt werden kann? Da die Mobilfunkabdeckung in den Bergen nicht überall besonders gut ist käme hier Satellitenkommunikation in Betracht.
- **Lokalisieren von Opfern:** Da die im Moment eingesetzte Technik der Lokalisierung von Opfern mittels LVS-Geräten auf der Basis von Langwellen sehr schnell und zuverlässig funktioniert, besteht hier kein Verbesserungsbedarf. Lediglich, wenn Opfer kein LVS-Gerät mit sich führen stellt sich das Auffinden schwierig dar. Hunde sind in der Lage auch solche Opfer zu lokalisieren. In Anlehnung an die vierbeinigen Helfer wäre es vorstellbar, dass spezielle Suchroboter, die Geruchssensoren eingebaut haben, eingesetzt werden. Vielleicht könnte man Geruchssensoren aber auch direkt in zukünftige LVS-Geräte einbauen. Inwieweit Geruchssensoren aber in der Lage sind Verschüttete aufzuspüren muss noch geklärt werden. Daneben könnten, unterstützend für die schnelle Lokalisierung, auch Infrarotsensoren in Form von Wärmebildkameras eingesetzt werden. Derzeit eingesetzte Geräte schaffen es jedoch nicht durch Schneedecken die größer als 5cm sind hindurchzumessen. Dadurch scheidet der Einsatz von Infrarotsensoren aus.
- **Ermitteln von Vitalwerten:** Für die Zukunft wäre es hier denkbar, dass neben der reinen Lokalisierung auch eine objektive Priorisierung vorgenommen wird. Dazu könnten Vitalfunktionen wie die Herzfrequenz, der Sauerstoffgehalt im Blut oder die Körpertemperatur erfasst und ausgewertet werden. Die Anzeige der Herzschläge auf einem LVS-Gerät könnte die Retter zusätzlich motivieren auch unter schwierigen Bedingungen weiter zu suchen. Die messtechnische Erfassung von Herzfrequenz und Gehalt des Sauerstoffs im Blut mittels Pulsometer stellt keine besonderen Anforderungen dar. Allerdings ist bei der Messung der Körpertemperatur darauf zu achten, dass sich,

bei einer möglichen Unterkühlung, die Restwärme auf den Körperkern mit den lebenswichtigen Organen zurückzieht. Eine Messung der Hauttemperatur an Händen oder Füßen kann dadurch von der relevanten Temperatur im Körperinneren stark abweichen. Diesem Problem könnte man beispielsweise durch das Schlucken eines tablettengroßen Thermometers entgegenwirken, das dann die Messwerte mittels Bluetooth oder Zigbee drahtlos nach außen sendet. Allerdings ist zu bezweifeln, dass der dadurch entstehende Nutzen im Verhältnis zu den mit dem Schlucken verbundenen Unannehmlichkeiten gerechtfertigt ist. Die Mehrzahl der Menschen würde es wohl ablehnen ein Thermometer zu schlucken. Ebenfalls in die Priorisierung könnten Daten einfließen, die angeben ob einer Person noch genügend Atemluft zur Verfügung steht. Dazu kämen spezielle Sauerstoffsensoren, sogenannte Lambdasonden, in Frage. Funktioniert dieses System, so wäre es möglich die Anzahl der Opfer zu minimieren.

- **Kommunikation** Um zu verhindern, dass mehrere Helfer nach der selben Person suchen, könnten sich die Geräte untereinander absprechen und so die Opfer den Rettern zuteilen. Dafür müsste ein für alle Geräte geltender, einheitlicher Standard zur Absprache geschaffen werden. Die prinzipielle Funktechnik von LVS-Geräten könnte aber beibehalten werden. Des Weiteren könnten die gemessenen Vitalwerte an professionelle Helfer übermittelt werden, die daraufhin den weniger geschulten Personen Anweisung zur ersten Hilfe geben könnten. Diese Funktion könnte vielleicht auch direkt vom LVS-Gerät übernommen werden, wodurch nur in komplizierten Fällen auf den Rat der Profis zurückgegriffen werden müsste. Dazu wäre es möglich Satellitenkommunikation zu nutzen.

Tatsächlicher Einsatz von Sensoren und Kommunikationstechniken zur Lawinenrettung: Nachfolgend werden einige ausgewählte Sensor- und Kommunikationstechnologien vorgestellt, die mögliche Lösungsansätze zu obigen Problemstellungen liefern:

- **Lokalisieren von Opfern:** Geruchslokalisierung könnte eine Möglichkeit bieten, verschüttete Opfer zu lokalisieren. Dies funktioniert zudem auch dann, wenn die betroffenen Personen nicht mit LVS-Geräten ausgestattet sind. So versucht das Team um Thomas Lochmatter vom Ecole Polytechnique Fédérale de Lausanne (EPFL), mittels Schwärmen von Miniaturrobotern Geruchsquellen ausfindig zu machen [Lochmatter 07]. Obwohl der Einsatz von Geruchssensoren derzeit kaum Verbreitung findet, können es die Besten von ihnen bereits mit der Empfindlichkeit von Hundenasen aufnehmen. Dabei ist das Prinzip recht simpel. Eine Geruchsquelle verteilt stetig Moleküle an die Umgebung, die durch diverse Luftströmungen zerstreut werden. Will man an die Quelle eines Geruchs gelangen, so untersucht man mittels Geruchssensor an verschiedenen Stellen die Luft nach den zu findenden Molekülen und misst dabei den Wind mit einem Anemometer. Jedoch ist dies alles andere als trivial. Zwar nimmt man die großen und starken Strömungen als Wind wahr und diese tendieren auch dazu die Moleküle in

eine bestimmte Richtung zu bewegen, allerdings beeinflussen gerade kleine, kaum merkbare Verwirbelungen den Luftstrom so, dass er nicht mehr vorhersagbar ist. Dieses Phänomen lässt sich auch schön an einem rauchenden Kamin beobachten. Nicht genug, dass sich die Konzentration des Rauches kaum vorhersagen lässt, sind auch die Strömungsrichtungen äußerst inkonsistent und wechseln ständig. Um Geruchsströmen folgen zu können benutzt man, ähnlich wie es beispielsweise Motten tun, ein Zick-Zack - Muster. Hierbei versucht man zick-zack-artig gegen den Wind, entlang dem Strom, zu kreuzen und kommt dadurch der Geruchsquelle immer näher.

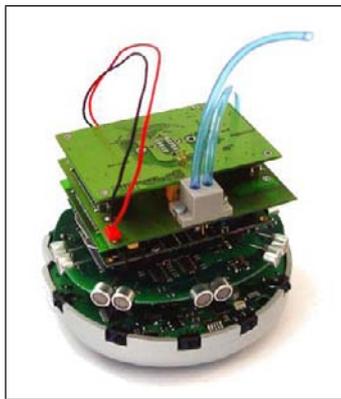


Abbildung 6. Khepera III Roboter ausgestattet mit Geruchssensor AirRobot der Bundeswehr und Anemometer zur Lokalisierung von Geruchsquellen. **Abbildung 7.** Aufklärungsdrohne

Eines der Hauptprobleme beim Einsatz von Robotern mit Geruchssensor ist die hohe Energieaufnahme. Um die Zeit bis zum ersten Ermitteln eines bestimmten Geruchsmoleküls zu verkürzen, setzen Forscher in diesem Gebiet auf eine Vielzahl kleiner Roboter. Zudem kann die Zeit weiter reduziert werden, wenn der Roboterschwarm untereinander kommuniziert. Lochmatter und sein Team verwendet dazu einen Khepera III - Roboter, der mit einem Geruchssensor und einem Anemometer ausgestattet ist (siehe Abb. 6). Dabei wird, ähnlich wie beim Menschen, die Luft mit einer Art Miniaturpumpe angesaugt und an den Geruchssensor weitergeleitet. Zur Kommunikation verfügt der Roboter zusätzlich über WLAN. Nach Lochmatter ist es durchaus denkbar, dass solche Roboter auch für zukünftige Such- und Bergungsmissionen von Verschütteten eingesetzt werden.

Da sich jedoch kleine Roboter in schwierigem Terrain nur ungenügend bewegen können und große Roboter wegen ihrem Gewicht ausscheiden⁶, kämen einzig fliegende Geruchsroboter in Betracht. Ähnlich den derzeit zu Aufklärungszwecken benutzten Quadroptern, wie etwa die Drohne AirRobot

⁶sie könnten höchstens von der Bergwacht eingesetzt werden.

der Bundeswehr (siehe Abb. 7), könnte man damit Geruchsquellen identifizieren und so Lawinopfer finden. Da Quadrocopter nicht besonders groß sind, wäre es eventuell sogar möglich sie bei Bergtouren in einem Rucksack mit sich zu führen um bei Bedarf schnell handeln zu können.



Abbildung 8. Zukünftiges LVS-Gerät wie es im A-Life Projekt eingesetzt wird, das neben der Position auch die relative Lage und den Vitalzustand der verschütteten Person anzeigt.

- **Ermitteln von Vitalwerten:** Bei einem Projekt der ETH Zürich mit dem Namen A-Life werden Ski- und Snowboardfahrer mit einigen Sensoren ausgestattet [Michahelles 03]. Mittels eines Pulsoximeters wird der Sauerstoffgehalt im Blut (SpO_4 in %) sowie die Herzfrequenz der jeweiligen Person gemessen. Als beste Positionierung des Messensors hat sich hier die Anbringung an einem Fußzeh erwiesen, da dieser durch den Schuh gut geschützt ist und so der Sensor auch bei einem Lawinenabgang in der Regel nicht verloren geht bzw. beschädigt wird. Wie bereits erwähnt zieht sich das Blut bei Unterkühlung aus den Extremitäten zurück, sodass das Messergebnis am Fuß von den tatsächlichen Werten abweicht. Allerdings ist Unterkühlung, zumindest in den ersten Stunden, nicht die Hauptbedrohung für das Opfer. Eine Lambdasonde, angebracht am Kragen der Jacke, misst die Konzentration von Sauerstoff in der Umgebung. Die Positionierung des Sensors am Kragen ist zwar nicht sehr sicher, hat sich aber bei Versuchen als beste, untersuchte Möglichkeit herausgestellt. Zusätzlich angebrachte Ultraschall-Transducer zur Distanzmessung geben weitere, nützliche Informationen über eine mögliche Höhle in der sich ein Verschütteter befindet. Als letzter Sensor wurde ein 3D - Beschleunigungsmesser direkt in das LVS-Geräte integriert, um so die relative Lage der verschütteten Person in der Lawine zu ermitteln. Das LVS-Gerät selbst besteht aus einem Personal Digital Assistant (PDA) (siehe Abb. 8). Die einzelnen Daten können entweder drahtlos, also mittels Technologien wie Bluetooth bzw. Zigbee, oder draht-

gebunden zwischen Sensoren und PDA ausgetauscht werden. Für letzteren Fall käme entweder ein Controller Area Network (CAN)-Bus System oder ein leitendes Gewebe⁷ in Frage. Der als LVS-Gerät eingesetzte PDA wertet die Messergebnisse aus und stellt sie dem Nutzer dar. Dabei werden nach Möglichkeit alle Daten doppelt abgeglichen. So wird beispielsweise Atemaktivität nur dann angenommen, wenn sowohl eine Luftblase im Schnee detektiert als auch die Sauerstoffsättigung im Blut mit über 80% gemessen wurde. Zudem überprüfen die Sensoren selbständig, ob sie korrekt montiert und die Messwerte gültig sind.

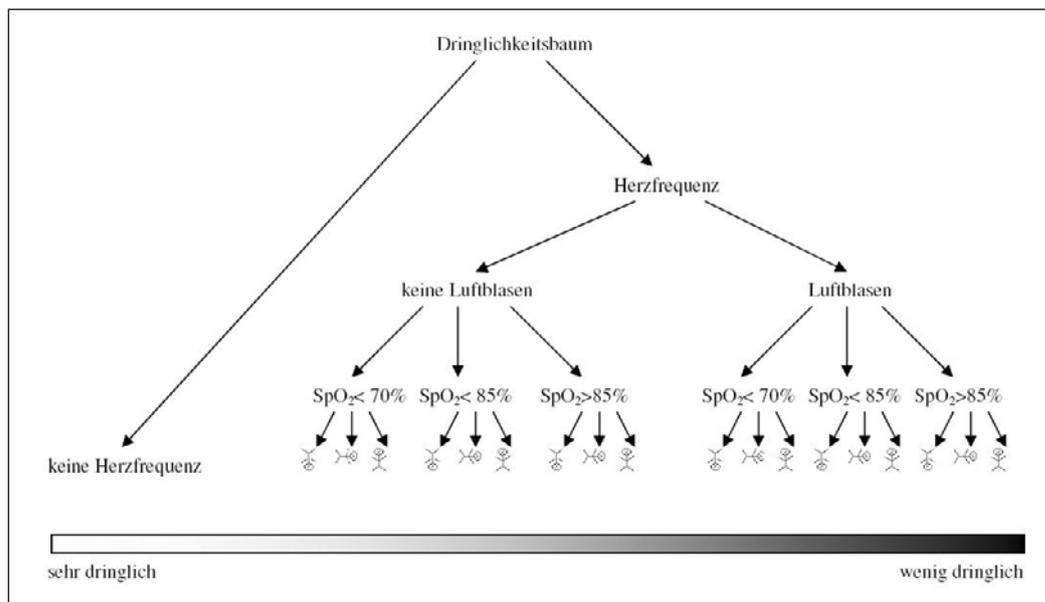


Abbildung 9. Dringlichkeitsbaum des A-Life Projekts, nachdem eine Priorisierung der Opfer vorgenommen wird.

Die Priorisierung der Dringlichkeit einer Rettung funktioniert wie folgt (siehe Abb. 9): Als Hauptkriterium wurde die Herzfrequenz genommen. Kann bei einer verschütteten Person kein Herzschlag gemessen werden, bekommt sie die höchste Dringlichkeit. Natürlich könnte man auch annehmen, dass es für diese Opfer bereits zu spät ist und man lieber nach einem noch lebenden Menschen suchen sollte. Diese Entscheidung muss letzten Endes der Retter selbst treffen. Als zweitwichtigstes Kriterium wurde das Vorhandensein von Luftblasen gewählt. Kann kein Sauerstoff in der Umgebung des Opfers detektiert werden, steigt die Dringlichkeit die Person zu retten. Drittes Kriterium ist die Sättigung des Bluts mit Sauerstoff. Schnelle Hilfe wird wichtiger, je weniger Sauerstoff sich im Blut befindet. Als letztes Merkmal wird die relative Lage des Opfers im Schnee herangezogen. Dabei ist die

⁷das Fraunhofer Institut hat ein Gewebe entwickelt, bei dem in den Stoff dünne Leitungen eingewoben sind.

Dringlichkeit die verschüttete Person zu bergen höher, je tiefer deren Kopf liegt. Die Funktionalität der Lokalisierung wurde von derzeit eingesetzten, handelsüblichen LVS-Geräten übernommen und arbeitet auf der Basis von Langwellen. Um die Rettung so objektiv wie möglich gestalten zu können, wurde auf Informationen wie Alter oder Geschlecht bewusst verzichtet. A-Life scheint die Lawinenrettung sinnvoll zu unterstützen, zumal der Mehrwert durch Funktionen, wie das automatische Absetzen eines Notrufs oder das Aufblasen eines Airbags, noch gesteigert werden kann.

- **Kommunikation:** Zur Kommunikation unter den einzelnen Geräten scheiden Techniken wie WLAN, Bluetooth oder Zigbee aus, da diese Schnee nicht durchdringen können. Die von derzeit eingesetzten LVS-Geräten benutzte 3-Antennen Technik zur Ortung anderer Geräte arbeitet im 457 kHz Bereich und schafft es so Signale auch durch Schnee zu empfangen. Allerdings unterstützt dieser Standard keine Übertragung von Daten. Zur Kompatibilität mit älteren Geräten wurde deshalb im A-Life Projekt ein zusätzliches Frequenzband zur Datenübertragung definiert. Die Kommunikation zwischen den suchenden LVS-Geräten wurde mittels WLAN realisiert. Da die WLAN-Technologie schon weit fortgeschritten ist und kein Schnee durchdrungen werden muss, schien dies die einfachste Variante.

Zum Absetzen eines Notrufs und der prinzipiellen Kommunikation mit möglicherweise weit entfernten Bergrettern wird derzeit GSM bzw. UMTS verwendet. Nach Bruno Jelk, Präsident der Internationale Kommission für Alpines Rettungswesen (IKAR), wurden 2007, 80 % der eingegangenen Notrufe über ein Mobiltelefon abgesetzt. Dazu ist die Mobilfunkabdeckung in den meisten Regionen ausreichend. Benutzt man ein Satellitentelefon, so hat man praktisch überall Empfang.

3.3 Erdbeben

Als letztes Szenario wird das eines Erdbebens angeschaut. Diese Katastrophen entstehen, wenn sich sogenannte tektonische Platten gegeneinander verschieben. An den Grenzen dieser Platten kommt es dann zu Erdbeben. Eines der heftigsten Erdbeben des letzten Jahrhunderts war das Beben vor Chile 1960. Mit einer Stärke von 9,6 auf der Richterskala entstand eine Bruchzone von mehr als 1500 km [Wikipedia 08]. Erdbeben können derzeit nicht vorhergesagt werden. Lediglich die Wahrscheinlichkeit mit der eine bestimmte Region von einem Beben getroffen wird kann berechnet werden.

Problematik: Bei einem Erdbeben sind meist große Bereiche mehr oder weniger stark von Schäden betroffen. Im Gegensatz zu den vorherigen Katastrophen fällt infolge der starken Erschütterung oft ein Großteil der vorhandenen Infrastruktur aus, was Bergungs- und Koordinierungsaufgaben erschwert. Da etwa Mobiltelefonnetze aufgrund der Zerstörung nicht mehr funktionieren oder einfach überlastet sind, ist die einzige Möglichkeit sich bemerkbar zu machen oft das Geben von Klopfzeichen oder lautes Rufen. Die größte Problematik ist

hier logistischer Natur. Notunterkünfte für Obdachlose müssen eingerichtet und die Grundversorgung mit Wasser und Nahrung sichergestellt werden. Wie man sieht, stellt die Bewältigung einer Erdbebenkatastrophe eine große Herausforderung dar. Das Retten von durch eingestürzter Häuser eingeschlossenen Personen ist ein schwieriger und langwieriger Prozess. Das liegt auch daran, dass Helfer nur wenig über Struktur und Stabilität der zusammengestürzten Bauten wissen und deshalb äußerst vorsichtig vorgehen müssen [ETHZ 08]. Auch kommt der Problematik, sich einen Überblick über die Situation zu verschaffen, eine ganz neue Bedeutung zu. Da Straßen und Brücken in der Nähe des Epizentrums oft nicht befahrbar sind, ist Spezialgerät zum Freiräumen und Bergen von Verschütteten notwendig. Das Ausmaß eines Erdbebens macht es anfangs unmöglich genaue Informationen über beispielsweise die Anzahl der Opfer zu ermitteln. Vielmehr steht im Vordergrund, den Umfang der Zerstörung grob einzugrenzen und schnellstmöglich die Kommunikation zur Koordinierung von Einsatzkräften wieder herzustellen bzw. neu aufzubauen.

Gefahren: Helfer und Verschüttete sind bei einem Erdbeben diversen Gefahren ausgesetzt, welche im Folgenden betrachtet werden:

- **Verletzungen während des Erdbebens:** Viele Menschen werden bereits beim Einsturz eines Gebäudes durch in sich zusammenstürzende Bauteile verletzt oder getötet.
- **Nachbeben:** Einer besonderen Bedrohung sind die Retter durch Nachbeben ausgesetzt. Bis zu mehreren tausend Nachbeben kann es infolge eines Erdbebens kommen, wobei die meisten von ihnen kaum wahrgenommen werden. Kommt es zu einem starken Nachbeben kann dies dazu führen, dass die Helfer selbst verschüttet oder getötet werden. Auch für vom ersten Beben eingeschlossene Menschen kann ein Nachbeben eine ernste Bedrohung darstellen.
- **Ausgehen der Atemluft:** Überlebende müssen mit eventuell begrenzten Sauerstoffreserven zurecht kommen. Ganz so kritisch wie bei einem Lawinenabgang ist dies aber nicht, da sich in einem eingestürzten Gebäude schon aufgrund der Bauart oft Hohlräume befinden, in denen ein Mensch einige Stunden und sogar Tage ausharren kann.
- **Brände:** Zudem treten auch durch Beben bedingte Brände auf. Feuerwehkräfte sind aufgrund der Vielzahl an Notfällen und der in Folge des Bebens eingeschränkten Mobilität oft nicht in der Lage schnell Hilfe zu leisten.
- **Seuchen:** Vor allem in wärmeren Gebieten entsteht nach einem größeren Erdbeben akute Seuchengefahr durch verwesende Körper.

Denkbarer Einsatz von Sensoren und Kommunikationstechniken zur Gefahrenminimierung: Da die Aufklärung der Lage und der Aufbau eines Kommunikationsnetzes diesem Szenario, im Unterschied zu den zuvor behandelten Katastrophen, eine ganz neue Tragweite beschert, werden diese Aspekte im Folgenden vertieft:

- **Aufklärung der Lage:** Um einen schnellen Überblick über ein von einem Erdbeben betroffenen Gebiet zu erlangen wäre es denkbar, dass fliegende Drohnen eingesetzt werden. Ähnlich wie im Militärbereich üblich könnten unbemannte, aus der ferne gesteuerte Flugzeuge das betroffene Gebiet überfliegen und mittels diverser Sensoren erfassen. Denkbar wäre hier der Einsatz von 3D-Laserscannern oder CCD-Sensoren. Diese könnten auch mittels Satellit das Katastrophengebiet aufklären und so sehr schnell eingesetzt werden. Eventuell wäre es möglich zur groben Aufklärung zuerst Satelliten einzusetzen und Drohnen nur dann punktuell zu verwenden, wenn noch präzisere Informationen gebraucht würden.
Daneben wäre es auch denkbar autonome oder ferngesteuerte Roboter zu benutzen. In Schwärmen wäre es möglich das Gebiet am Boden aufzuklären und so Informationen auch aus Gebäuden zu liefern, in die Drohnen oder Satelliten nicht hineinsehen können. Da in den ersten Stunden wohl weit mehr Hilfe gebraucht wird als Helfer vorhanden sind ist es unabdingbar, dass auch Roboter selbst Menschen detektieren und eventuell auf einer Karte einzeichnen. Denkbare Sensoren wären Infrarotsensoren, Mikrofone und CCD-Sensoren. Rettungsteams könnten dann gezielt diese Punkte abgehen und Opfer bergen. Fraglich ist jedoch, inwieweit die mechanischen Helfer fähig sind sich in schwierigem Terrain zu bewegen.
- **Aufbau eines Kommunikationsnetzes:** Neben der Information für die Koordinierung des Einsatzes wäre es ebenfalls sinnvoll, wenn auch die Rettungskräfte mit für sie wichtigen Informationen versorgt würden. Deshalb muss ein Kommunikationsnetz verfügbar sein. Da, wie bereits angedeutet, in Folge eines Erbebens davon auszugehen ist, dass herkömmliche Kommunikationssysteme nicht verfügbar sind, muss dieses neu aufgebaut werden.
Denkbar wäre es dabei, das im Umfeld des Großbrandszenarios angesprochene DUMBONET einzusetzen. Die hieraus resultierende Problematik, dass der Aufbau des Netzes 2-3 h dauern kann, würde sich bei einem Erdbeben wahrscheinlich auszahlen. Zumal das Eintreffen von Helfern und Bergungsgeräten ebenfalls seine Zeit braucht.

Tatsächlicher Einsatz von Sensoren und Kommunikationstechniken zur Gefahrenminimierung:

- **Aufklärung der Lage:** In einem Paper von Alexander Kleiner [Kleiner 04] beschreibt er und sein Team die Möglichkeit, mittels vieler mobiler Agenten ein großes Szenario aufzuklären. Jede Person ist dabei mit einer zentralen Einsatzleitung verbunden, wobei vom genauen Aufbau des Netzes abstrahiert wird. Der Leser erfährt lediglich, dass verschiedenste Kommunikationssysteme wie Handys oder Ad-hoc-Netze verbunden werden können. Mittels Wegpunkt-Nachrichten kann jeder Erkunder Informationen an die Einsatzleitung melden. Diese Nachrichten können das Erkennen einer Blockade oder das Auffinden eines Opfers sein aber auch die Information enthalten, dass ein bestimmtes Gebäude oder ein bestimmter Bereich aufgeklärt wurde. Um

die Nachrichten der aktuellen Position zuordnen zu können, sind alle Agenten mit GPS-Sensoren ausgerüstet. Ausgehend von einer initialen Karte, die etwa ein Satellitenbild, aufgenommen vor der Katastrophe, darstellen kann, werden die übermittelten Informationen automatisch in die Karte integriert. Dies kann dann mittels eines Head-Up-Displays vom einzelnen Retter betrachtet werden. So wird ein Graph erstellt der sichere Wege zu Opfern enthält und anzeigt welche Gebiete des Erdbebens bereits exploriert wurden und welche noch nicht.

Eine Arbeit von H.-P. Bähr und Kollegen [Bähr 04] benutzt Laserscanner die an ein Flugzeug montiert sind (siehe Abb. 10). Der Einsatz von CCD-Sensoren erwies sich dagegen in [Shibayama 08] als ungeeignet, da die Aufnahmen stark wetterabhängig sind und auch bei guter Sicht nur bei Tag gemacht werden können. Um die Abtastung einem Ort zuordnen zu können, wird mittels DGPS-Sensor die aktuelle Position des Flugzeugs ermittelt. Ziel ist es Schäden automatisch festzustellen. Dazu muss jedoch bereits ein Abbild des Gebiets vor dem Erdbeben vorliegen. Vergleicht man nun beide 3-D - Lasermodelle kann daraus die Veränderung, also die Zerstörung durch das Erdbeben, errechnet werden. Mit diesem Messverfahren können Schäden die mehr als 1 m² umfassen detektiert werden.

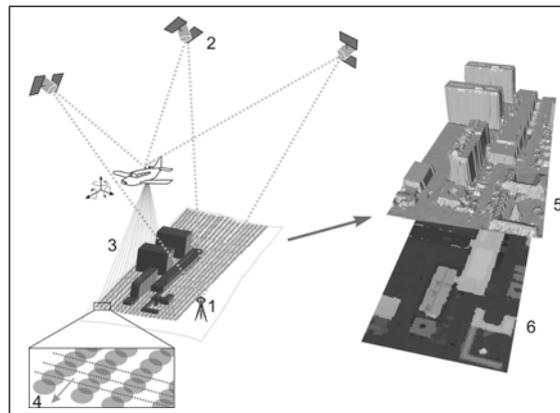


Abbildung 10. Prinzip des Laserscannens mit einem Flugzeug. 1+2: DGPS - Ausrüstung, 3: Lasermessung, 4: Scanmuster, 5+6: 3D - bzw. 2D - Modell des Scans.

Der Einsatz von UWB-Radios als Sensoren in Erdbebengebieten wird in einem weiteren interessanten Paper beschrieben [Lee 07]. Da Rettungskräfte bei zusammengestürzten Gebäuden häufig keine Informationen über deren Struktur haben, müssen sie bei der Bergung äußerst vorsichtig sein. Es kann passieren, dass beim Entfernen der falschen Gegenstände das Haus weiter in sich zusammenbricht, was fatale Folgen haben kann. Um dies zu verhindern werden UWB-Radios schon beim Bau in die Wände integriert. Ein Querschnitt eines eingestürzten Hauses ist in Abb. 11 dargestellt. Zwar werden eine Vielzahl an Sensoren schon beim Einsturz des Hauses zerstört, jedoch

kann Information über die innere Struktur, auch wenn sie nicht vollständig ist, hilfreich für die Retter sein. Zum „Scannen“ der Umgebung übermitteln UWB-Sensornodes ein Signal mit einer festgelegten Signalstärke. Jeder Sensorknoten besitzt zwei UWB-Empfänger die in definiertem Abstand angebracht sind. Zudem wird angenommen, dass für ein Bauwerk typische Baustoffe benutzt werden⁸. Mit dem Wissen über die Stärke des ausgesendeten Signals und den zwei Empfangsantennen, lässt sich mit einer bestimmten Wahrscheinlichkeit berechnen welches Material mit welcher Materialstärke zwischen zwei Knoten liegt. Das Projekt befindet sich derzeit noch in der Anfangsphase hat aber schon eine Erkennungsrate zwischen 98 und 100 %. Für die weitere Forschung ist geplant die Sensorknoten zusätzlich mit Ultraschall-Transducern zu erweitern.

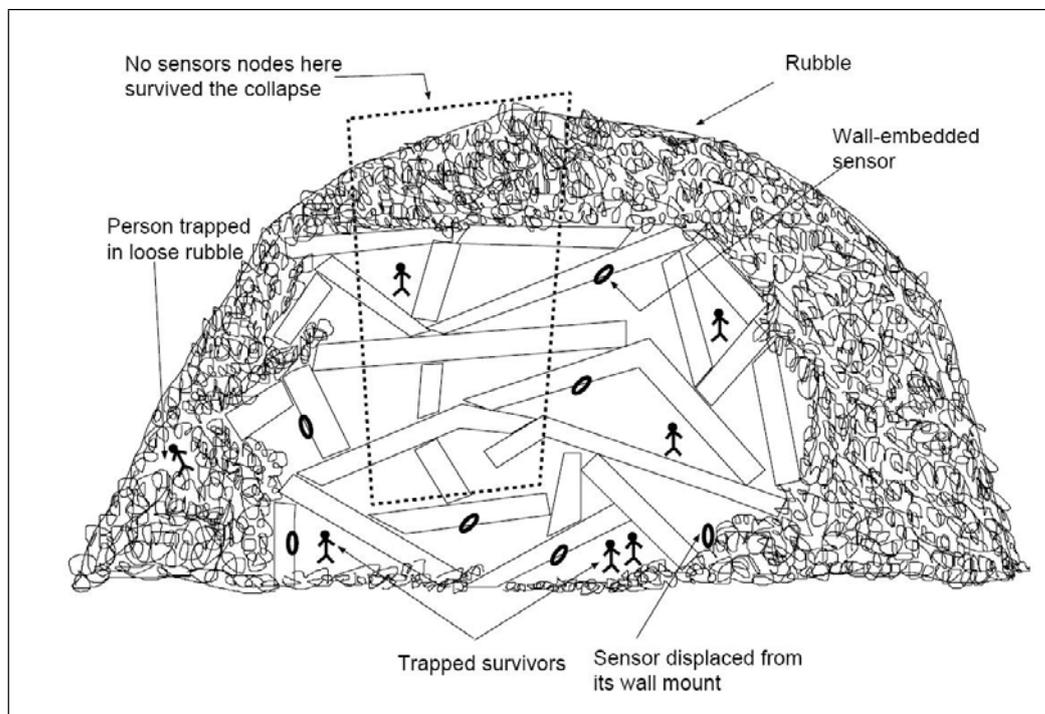


Abbildung 11. Ausschnitt eines eingestürzten Gebäudes, das mit UWB-Sensoren ausgestattet ist.

- **Aufbau eines Kommunikationsnetzes:** Neben dem Einsatz von DUMBONET könnten Funkmodule etwa im 434 MHz Bereich genutzt werden. Sie schaffen Reichweiten von 600 Meter und können mit bis zu 10 Relaisstationen eine Reichweite von ca. 7 km erreichen. Geräte wie das MU-1, die eine solche Funktechnik unterstützen, haben zudem eine Anzeige für den Empfangspegel ähnlich dem am Handy. Ein System dieser Art wird beispiels-

⁸Im Paper wurden dafür Beton, Stahlbeton, Trockenmauer und Holz sowie Kombinationen aus diesen untersucht und ihre Charakteristiken ausgewertet.

weise zur Überwachung von Erdbeben eingesetzt. Da es sich um einen frei verfügbaren Frequenzbereich handelt, könnte die Kommunikation allerdings im Ernstfall gestört werden.

4 Fazit

Katastrophen jeglicher Art werden sich wohl auch in Zukunft nicht vermeiden lassen. Die Forschung scheint sich der Problematik aber bewusst und liefert einige vielversprechende Ansätze um die im Einsatz auftretenden Probleme mit Hilfe von Sensoren und Kommunikationseinheiten zu verringern. Eine der größten Herausforderungen stellt dabei die schnelle Aufklärung der Lage dar. Mehr Informationen bedeutet sowohl mehr Schutz für die Rettungskräfte als auch eine höhere Überlebenschance für die Opfer. Sensoren sind in der Lage diese zusätzlichen Informationen zu erbringen, sodass sie mittels moderner Kommunikationstechniken in Echtzeit verteilt werden können. Die perfekte Lösung für die Vielzahl an Fragestellungen ist sicherlich noch nicht überall gefunden aber man ist auf einem guten Weg.

Weiterführende Literatur

- [Bähr 04] H.-P. Bähr, A. Hering Coelho, J. Leebmann, E. Steinle, D. Tóvári. Geospatial Data Acquisition by Advanced Sensors in Disaster Environments. 2004.
- [Bixby 07] Lyn Bixby. Firefighter Safety Spurs Interest In Robots. Fire Apparatus & Emergency Equipment, 12, 2007.
- [Brushlinsky 06] N.N. Brushlinsky, J.R. Hall, S.V. Sokolov, P. Wagner. World Fire Statistics. Centre of Fire Statistics of CTIF, 11, 2006.
- [Eric 08] Eric. Intelligent clothing for the safety of Firefighters. Online-Quelle. < <http://www.talk2myshirt.com/blog/archives/390> >.
- [ETHZ 08] ETHZ. Erdbeben: Häufig gestellte Fragen. Online-Quelle. < <http://www.seismo.ethz.ch/info/faq/faqde.htm> >.
- [Hinck 06] Steffen Hinck. RESCUE - wearable computer in desaster szenarien. 2006.
- [Hinck 07] Steffen Hinck. Einsatz von Wearable Computing in Desaster Szenarien - Seminararbeit. 2007.
- [Jacoff 03] Adam Jacoff, Brian Weiss, Elena Messina. Evolution of a Performance Metric for Urban Search and Rescue Robots. 2003.
- [Kanchanasut 07] Kanchana Kanchanasut, Apinun Tunpan, Mohammad Abdul Awal, Dwijendra Kumar Das, Thirapon Wongsaardsakul, Yasuo Tsuchimoto. A Multimedia Communication System for Collaborative Emergency Response Operation in Disaster-affected Areas. 2007.
- [Klann 07] Markus Klann, Till Riedel, Hans Gellersen, et al. LifeNet: an Ad-hoc Sensor Network and Wearable System to Provide Firefighters with Navigation Support. 2007.
- [Kleiner 04] Alexander Kleiner, Nils Behrens, Holger Kenn. Wearable Computing meets Multiagent Systems: A real-word interface for the RoboCupRescue simulation platform. 2004.
- [Lee 07] Jeongeun Julie Lee, Suresh Singh. Using UWB Radios as Sensors for Disaster Recovery. 2007.
- [Lochmatter 07] Thomas Lochmatter, Xavier Raemy, Alcherio Martinoli. Odor Source Localization with Mobile Robots. Bulletin of the Swiss Society for Automatic Control, 2007.

- [Michahelles 03] Florian Michahelles, Timo Ahonen, Bernt Schiele. A-Life - Increasing Survival Chances in Avalanches by Wearable Sensors. 2003.
- [Scheuren 07] J-M. Scheuren, O. le Polain de Waroux, R. Below, D. Guha-Sapir, S. Ponserre. Annual Disaster Statistical Review. 2007.
- [Schulenburg 07] Mathias Schulenburg. Forschung für die zivile Sicherheit. Bundesministerium für Bildung und Forschung (BMBF), 2007.
- [Shibayama 08] Takashi Shibayama, Takashi Nonaka, Susumu Takagishi, Tada-shi Sasagawa. Disaster Information Acquisition System using Multi-Platform Remote Sensing Technology. 2008.
- [Wikipedia 08] Wikipedia. Die Freie Enzyklopädie (deutsche Version). Online-Quelle. < <http://de.wikipedia.org/wiki/Hauptseite> >.

CAN	Controller Area Network
CCD	Charge-Coupled Device
DGPS	Differential Global Positioning System
DUMBONET	Digital Ubiquitous Mobile Broadband OLSR Net
EADS	European Aeronautic Defence and Space Company
EPFL	Ecole Polytechnique Fédérale de Lausanne
GEO	Geostationary Earth Orbit
GPRS	General Packet Radio Service
GPS	Global Positioning System
GSM	Global System for Mobile Communications
HSDPA	High Speed Downlink Packet Access
IKAR	Internationale Kommission für Alpines Rettungswesen
LED	Light Emitting Diode
LEO	Low Earth Orbit
LVS	Lawinen-Verschütteten-Suchgeräte
MEO	Medium Earth Orbit
OLSR	Optimized Link State Routing
PAN	Personal Network
PDA	Personal Digital Assistant
PSA	Persönliche Schutzausrüstung
UMTS	Universal Mobile Telecommunications System
UWB	Ultra Wideband Communication
VPN	Virtual Private Network
WLAN	Wireless Local Area Network
WMAN	Wireless Metropolitan Area Networks