



Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Projektausarbeitung

Martin Stein

RFID-Tracking

Betreuender Prüfer: Prof. Dr Gunter Klemke

Martin Stein

Thema der Projektausarbeitung

RFID-Tracking

Stichworte

Tracking, RFID, Radio Frequency Identification, EPC, Electronic Product Code, EPCglobal Network, Wireless Sensor Network, Ferienclub

Kurzzusammenfassung

Im Wintersemester 2005/2006 wurde im Master-Studiengang Informatik an der HAW-Hamburg ein Projekt zum Thema "Ferienclub" durchgeführt. Ein Teilprojekt befasste sich hierbei mit dem Thema RFID-Tracking. Dieses Papier ist die Projektausarbeitung eben dieses Teilprojektes und gibt eine kurze Übersicht über die eingeschlagenen Lösungswege und erreichten Ziele innerhalb des Teilprojekt RFID-Tracking.

Inhaltsverzeichnis

1	Standardisierungsbemühungen der Auto-ID Center und von EPCglobal	5
2	CAD-Daten für ein ad-hoc Tracking	8
3	Lokale Eventfilterung	9
4	Fremde Datenquellen	13
5	Mobile Lesegeräte	14
6	Übergeordnete Tracking-Architektur	15
7	Wegvorhersage	16
8	Versuchs-Implementation	20

Das Teilprojekt „RFID-Tracking“ als ein einzelner Bestandteil des Master-Projektes „Ferienclub“ hatte zur Aufgabe die Bewegungen von Gästen, Mitarbeitern und ausleihbarem Club-Inventar auf dem Clubgelände zu verfolgen. Die gesammelten Tracking-Informationen sollten anderen Teilgruppen des Ferienclubs zur Verfügung gestellt werden. Für die Bewegungsverfolgung sollten die zu trackenden Objekte mit RFID-Transpondern ausgestattet werden. Für den ersten Entwicklungsschritt waren ausschließlich stationäre Lesegeräte vorgesehen. Hierbei sollten die einzelnen RFID-Lesegerätstationen versuchen, bei „ungeplanter“ Ausbringung im zu überwachenden Gebiet selbstständig ein Funknetzwerk und vor allem ein Tracking-System zu bilden (vergleichbar mit einem Ad-hoc Wireless-Sensor-Netzwerk¹). Für einen etwaigen zweiten Entwicklungsschritt sollte die Möglichkeit zur Integration von zusätzlichen in diesem Fall dann mobilen Lesegeräten vorgesehen werden. Die bereits angesprochene Analogie zu Wireless-Sensor-Netzwerken ließ auch gleich den Wunsch aufkommen, Teile des Tracking-Algorithmusses direkt ins Netzwerk zu verlagern. Dieses Vorgehen soll bewirken, dass nicht mehr alle erfassten Sensordaten ungefiltert an ein zentrales Tracking-System übertragen werden müssen, sondern dass bereits einzelne Netzwerkknoten und Netzwerkknotenverbünde versuchen können, den zentralen Tracking-Server von unnötigen Daten zu entlasten.

Dieses Papier ist im Folgenden in acht Kapitel gegliedert. Als erstes wird beschrieben, in wieweit das entwickelte System Nutzen aus den Standardisierungsbemühungen im Hard- und Softwarebereich rund um das *EPCglobal Network* ziehen konnte. Im darauf folgenden

¹ Siehe auch *Mobile Ad-hoc Networks und Wireless-Sensor-Networks*[Stein 2006]

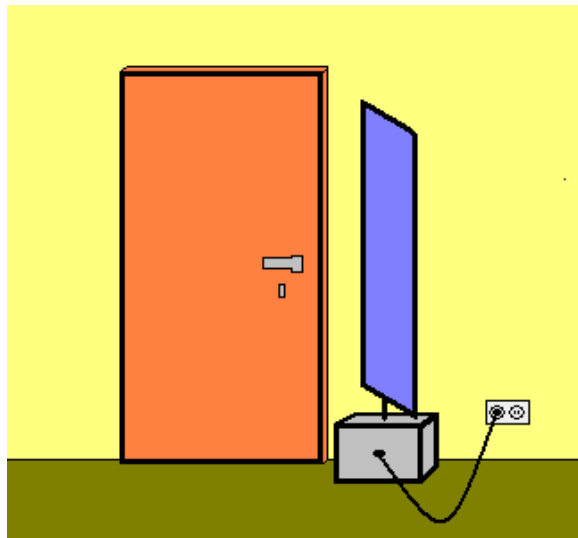


Abbildung 1: Der Wunsch - RFID-Lesegeräte einfach nur an eine 230-Volt-Steckdose neben einer Tür stecken und das gesamte Tracking-System baut sich per Funk von alleine auf

Kapitel wird dargelegt, wie versucht werden sollte, mit Hilfe von Gebäude-CAD-Daten und einer automatischen Positionsbestimmung der RFID-Lesegerätstationen ein Tracking-System ad-hoc zu formieren. Das dritte Kapitel beschäftigt sich mit einem Lösungsansatz für eine lokal-kooperative Eventfilterung zwischen RFID-Lesegerätstationen. Kapitel vier und fünf beschreiben daraufhin, wie der Lösungsansatz aus Kapitel drei durch mobile RFID-Lesegeräte und „artfremde“ Event-Quellen (andere Dinge als RFID-Sichtungen) erweitert werden könnte. Kapitel sechs beschreibt die übergeordnete Tracking-Architektur über dem bis hierher dann vorgestellten lokalen Trackingssystem. In Kapitel sieben wird sodann noch eine einfache Wegvorhersage vorgestellt, wobei am Rande noch ein Vergleich zu Markow-Ketten vorgenommen wird. Kapitel acht skizziert noch grob den Aufbau der testweisen Implementation des Systems².

Sollte der Leser der Meinung sein, dass seine Kenntnisse zu den Themen *RFID*, *EP-Cglobal Network* und *Wireless-Sensor-Networks* eine kleine Auffrischung bedürfen, wird empfohlen vorab die vom Autor dieses Papiers ebenfalls veröffentlichten Papiere *Mit RFID-Technologie zum Internet der Dinge*[Stein 2005b] und *Mobile Ad-hoc Networks und Wireless-Sensor-Networks*[Stein 2006] zu lesen. Zudem kann an dieser Stelle auch das BITKOM White Paper *RFID Technologie, Systeme und Anwendungen* als kompakte Einstiegshilfe empfohlen werden (deutsch: [BITKOM 2005a] englisch: [BITKOM 2005b]).

²Die hierauf folgenden eigentlich noch geplanten drei Kapitel, mussten auf Grund der festgesetzten Seitenzahlbeschränkung leider gänzlich entfallen. So bleiben die Themen „Lösungsansatz für eine verteilte gelöste Wegvorhersage“, „Allgemeinen Erkenntnisse aus den Verteilungsüberlegungen“ und „Kritische Auseinander mit den gewonnenen Erkenntnissen und der darauf folgenden teilweisen Software-Realisierung“ außen vor

Zu beachten ist ferner, dass dieses Papier zwingend in Kombination mit den Folien des hiermit direkt im Zusammenhang stehenden Projekt-Vortrages zu lesen ist (Folien siehe Anhang).

1 Standardisierungsbemühungen der Auto-ID Center und von EPCglobal

Seit einiger Zeit werden die Worte *RFID*, *Tracking* und *EPCglobal Network* immer wieder in nur einem einzigen Atemzug genannt. In logischer Konsequenz daraus war im Semester vor diesem Projekt angedacht worden, zu Beginn des Projektes zu untersuchen, inwieweit man einzelne EPC-Komponenten sowie EPC-“Standardsoftware“ für dieses Tracking-Vorhaben einsetzen könnte. Leider zeigten die im Semester zuvor gefundenen und sich mit dem EPC-Thema befassenden OpenSource-Projekte nicht die erhofften Fortschritte und konnten daher im Projekt noch nicht genutzt werden³. Eine genauere Beschäftigung mit der EPCglobal Network Architektur machte zudem klar, dass sich die gewünschte kooperative Daten-Verarbeitung (bzw. Event-Filterung) zwischen den RFID-Lesegerät-Knoten nur umständlich realisieren ließe.

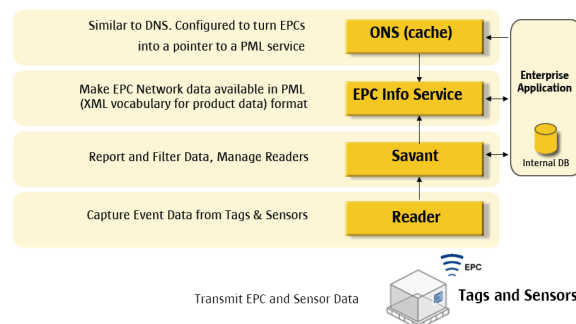


Abbildung 2: EPCglobal Network Architektur

In der EPCglobal Network Architektur (siehe Abb. 2) lassen sich einfache Filter direkt in den einzelnen RFID-Lesegeräten finden. Hier können zum Beispiel mehrere gleiche Lese-Events innerhalb weniger Sekunden zu einem einzelnen Lese-Event zusammengefasst werden. Umfangreichere Event-Filterungen finden in den Savants statt. Hier kann zum Beispiel mit Hilfe von zusätzlichen Daten aus einer Datenbank ein Pulk von Transponder-IDs als eine einzelne zusammengehörige Lieferung erkannt werden, sodass an übergeordnete Systeme

³RadioActive Neutrino, Fusion und Graviton <http://www.radioactivehq.org> und xcEPC (eXtreme Commerce EPC) Network Core for Java <http://www.refactored-networks.com/downloads.html>

nur noch ein einzelnes Lese-Event über eine Lieferungs-ID erfolgen muss. Dass sich mehrere Lesegeräte in direkter Nachbarschaft darüber verständigen könnten, welche Events sie vielleicht nicht an den Savant schicken müssen, ist nicht direkt vorgesehen.

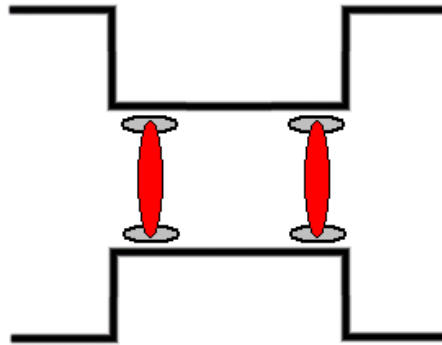


Abbildung 3: RFID-Lesegerät-Schleuse trennt zwei Räume

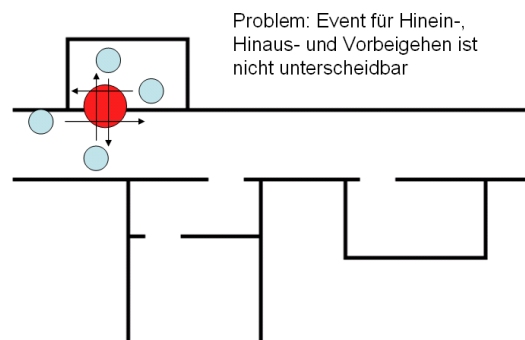


Abbildung 4: Event-Unklarheit

Ein Grund hierfür dürfte unter anderem darin liegen, dass in den Logistikprozessen, auf welche das EPC-System abzielt, quasi gar keine Unklarheiten in den Lese-Events enthalten sind. Anders bei den Lese-Events in dem hier zu entwickelnden Ferienclub-System, wo es eine Bewegungsrichtungsunsicherheit gibt. Um solche Unsicherheiten zu vermeiden, wird in der Logistik darauf geachtet, dass es dedizierte Eingangsbereiche mit eigenen RFID-Lesegerät-Toren für zum Beispiel die Einlieferung von Waren in eine Lagerhalle gibt und klar davon unterscheidbare Ausgangsbereiche mit wiederum eigenen RFID-Lesegerät-Toren. Sollte diese Trennung so nicht möglich sein, werden beispielsweise Lesegerät-Tunnel bestehend aus zwei hintereinander liegenden RFID-Lesegerät-Toren aufgestellt (siehe Abb. 3)

oder mehrere Lichtschranken detektieren die Bewegungsrichtung der die Waren transportierenden Gabelstapler. Wie bereits erwähnt, lässt unserer Ferienclub-Umgebung normalerweise keine genauen Schlüsse auf die Bewegungsrichtung der detektierten RFID-Transponder zu (siehe Abb. 4), sodass typischerweise erst zwei aufeinanderfolgende und unterschiedliche Lese-Ereignisse zu einer neuen Information führen (mehr dazu in Kapitel 3). Aber nicht nur die Möglichkeit einer lokal-kooperativen Event-Filterung machte eine Entwicklung einer eigenen Tracking-Architektur interessant und erstrebenswert. Auch die Überlegung, sich während dieses Masterprojektes zumindest erste Gedanken darüber zu machen, wie man eine Wegvorhersage verteilt im Netzwerk lösen könnte, schien eine interessante Herausforderung zu sein⁴.

Das Fehlen von frei verfügbarer EPC-Software und die für unser Vorhaben nicht ganz geeignete Architektur des EPCglobal Network Systems, sowie ein gewisser akademischer „Spieltrieb“, der aus dem Wunsch entsprang, soviel wie möglich verteilt zu lösen und so nah wie möglich am Ort der „Sensormessung“ die „Sensordaten“ zu verarbeiten, führte schließlich zu der Entscheidung eine ganz eigene Architektur zu entwerfen.

Im Gegensatz zu der Frage nach der Software-Architektur wurde bei der Suche nach geeigneter RFID-Hardware jedoch besonderer Wert auf Kompatibilität zu EPC gelegt. Da sich bereits einige große Einzelhandels- und Logistikunternehmen sowie das amerikanische Verteidigungsministerium dafür entschieden haben, für ihre Pilotprojekte EPC-kompatible Hardware im UHF-Frequenzbereich einzusetzen, ist die Wahrscheinlichkeit sehr groß, dass diese Geräte in naher Zukunft eine große Marktdurchdringung erlangen werden. Die zu erwartende Massenproduktion solcher Hardware sollte sich nicht nur positiv auf die zukünftig möglicherweise nötig werdende Wiederbeschaffung von ausgefallener Hardware auswirken, sondern sollte auch die Kosten für eine zukünftig denkbare Erweiterung eines vielleicht mal existierenden Produktivsystems günstig beeinflussen.

Für die stationären Lesegeräte kamen nach eingehender Produktrecherche zwei UHF-Geräte in die engere Wahl: Der ALR-8780⁵ der Firma Alien Technology und der OBID i-scan ISC.LRU1000⁶ der Firma FEIG. Beide Lesegeräte entsprechen der ETSI-Spezifikation EN 302 208 und dürfen somit mit der seit neusten in Deutschland maximal zulässigen Sendeleistung von 2 Watt betrieben werden (vorher max. 0,5 Watt). Serienmäßig können bis zu vier Antennen an die einzelnen Lesegeräte angeschlossen werden, da sie bereits einen vierfachen Antennen-Multiplexer besitzen. Zudem gibt es mittlerweile für beide Lesegeräte Firmwareupdates zur Unterstützung von EPC Class 1 Gen 2 Transpondern. Einige Unterschiede in den Details der Lesegeräte, welche an dieser Stelle leider nicht ausführlich besprochen werden können, führten schließlich jedoch zu einer Favorisierung des Lesegerätes von Alien Technology. Im Falle der mobilen Lesegeräte fiel die Entscheidung auf eine UHF EPC

⁴In dieser Ausarbeitung ist die verteilt Wegvorhersage nicht beschrieben

⁵<http://www.alientechnology.com/products/alr8780.php>

⁶http://www.feig.de/index.php?option=com_content&task=view&id=66&Itemid=119&lang=de

PCMCIA-Karte von WJ Communications. Die MPR 5000 RFID-PCMCIA-Card gibt es bisher leider nur mit FCC-Zulassung. Auf Nachfrage beim schweizer Distributor wurde mitgeteilt, dass mit einer europäischen Version leider nicht vor dem 2Q 2006 zu rechnen ist.

Da die stationären Lesegeräte bisher noch sehr kostenintensiv sind (jeweils um die 3000 Euro) und auch eine mobile RFID-Lesegeräte Lösung für zukünftige Hochschul-Projekte gesucht wurde, ging die Recherche vorerst weiter und führte zum CPR.PR50 der Firma Softwork. Das CPR.PR50 Lesegerät arbeitet auf der für EPC-Anwendungen eher unüblichen Frequenz von 13,56 MHz und ist durch seine relativ geringe Reichweite von nur einigen Zentimetern für viele Anwendungen auch schlicht ungeeignet. Für dieses Lesegerät spricht aber der geringe Komplettpreis von unter 400 Euro und die Treiberkompatibilität zum bereits oben erwähnten stationären RFID-Lesegerät OBID i-scan ISC.LRU1000 von FEIG. Diese Kompatibilität kommt daher, dass der italienische Hersteller Softwork im inneren des CPR.PR50 das RFID-Lesemodul OBID classic-pro ID CPR.M02 der in Deutschland ansässigen Firma FEIG verwendet. Besonders interessant für zukünftige Hochschul-Projekte dürfte sein, dass der CPR.PR50 nicht nur kompatibel zu ISO 15693 bzw. EPC⁷ Transpondern ist, sondern ebenso zu den bereits weit verbreiteten Transpondern nach ISO 14443 A und B. Zudem bietet das Lesegerät über seinen SAM (Security Access Module) Anschluss auch eine Erweiterungsmöglichkeit für ausgewählte kryptographisch arbeitende Transponder an.

2 CAD-Daten für ein ad-hoc Tracking

Nun aber weg von der Hardware wieder zurück zu den Softwareaspekten des hier beschriebenen Tracking-Projektes. Ein Wunsch war es, ein Tracking-System zu entwickeln, welches wirklich von jedem aufgestellt und betrieben werden kann. Dazu wäre es im Idealfall für den Kunden nur nötig eine genügende Anzahl von RFID-Lesegerätstationen und einen zentralen Tracking-Server einzukaufen, sowie eine CAD-Zeichnung des Gebäudes, in dem das Tracking stattfinden soll, zu organisieren. Anschließend müsste der Kunde seine eingekauften RFID-Lesegerätstationen nur noch an strategisch wichtigen Türen und Durchgängen an das 230 Volt Stromnetz anschließen (siehe Abb. 1), um anschließend darauf zu warten, dass sich das Tracking-System selbst konfiguriert. Dafür müssten dann noch kurzzeitig drei präzise GPS-Empfänger außerhalb des Gebäudes aufgestellt werden, damit die RFID-Lesegerätstationen nach dem Aufstellen ihre absolute Position bestimmen könnten (siehe dazu relative und absolute Positionsbestimmung mit Hilfe von UWB in [Stein 2006]⁸. Zum

⁷EPC HF - nicht UHF Transponder

⁸Anmerkung: Da UWB erstens noch keine Zulassung in Europa hat und zweitens noch keine Consumer-Produkte erhältlich sind, wurde als eine mögliche Alternative für eine Mesh- Vernetzung im Ferienclub (aber ohne genaue Positionsbestimmung) der VirtualWiFi-Treiber von Microsoft Research gefunden <http://research.microsoft.com/netres/projects/virtualwifi/default.htm>

Ablauf, wie das Tracking-System selbständig eine Raumaufteilung vornehmen sollte, siehe Vortragsfolie drei bis sieben (siehe Anhang).

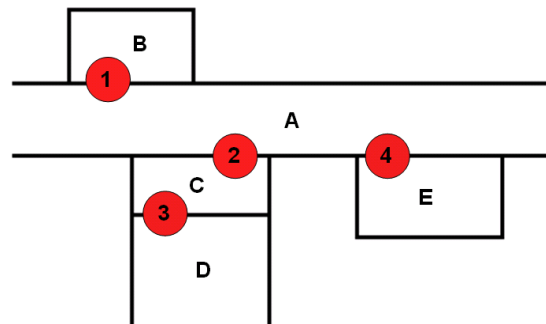


Abbildung 5: Unterteilung des Gebäudes in Räume

Ein großes Problem stellten dabei die CAD-Daten dar, welche idealerweise mit den Teilprojekten *3D-Visualisierung* und *Location Based Services* gemeinsam genutzt werden sollten. Als Datenformat wollten wir gerne SVG verwenden. Leider unterstützte das anfänglich zur Verfügung stehende CAD-Programm ausschließlich schwer handhabbare AutoCAD-Formate, wobei uns aber ein AutoCAD zu SVG Konverter vorlag. Leider musste nach der Einarbeitung in das CAD-Programm festgestellt werden, dass die AutoCAD-Formate offensichtlich nicht 100%ig standardisiert sind, sodass eine Konvertierung misslang. Ein danach ausprobiertes CAD-Programm konnte zwar direkt SVG-Dateien ausgeben, doch wurde bei der Durchsicht der resultierenden SVG-Dateien festgestellt, dass das Programm offensichtlich u.a. einfache Linien in viele kurze Linien aufteilte, sodass die SVG-Dateien quasi in Bitmaps entarteten. Um nicht noch mehr Zeit für die Realisierung dieser Komponente des Tracking-Systems in Anspruch zu nehmen, wurde entschieden, an dieser Stelle abzubrechen und für die testweise Implementierung des Tracking-Systems die Konfiguration des Tracking-Netzwerks per Hand vorzunehmen.

3 Lokale Eventfilterung

Das bereits in Kapitel 1 angesprochene Problem der Event-Unterscheidung (siehe Abb. 4) führte nach einigen Entwicklungsschritten zu einer Lösung, welche in den Vortragsfolien 13 bis 25 in Bildern dargestellt wird (siehe Anhang).

Im Folgenden wird erklärt, was im Sequenzdiagramm in Abbildung 8 im Einzelnen zu sehen ist:

1.) Das RFID-Lesegerät L1, welches zwischen Raumknoten A und B aufgestellt ist (siehe Abb. 6) sendet bei einer Transponder-Sichtung die gelesene ID an Raumknoten A (siehe Abb. 7 Zeile eins und Abb. 8 Sequenz eins). Angenommen die gesichtete ID ist zu Anfang

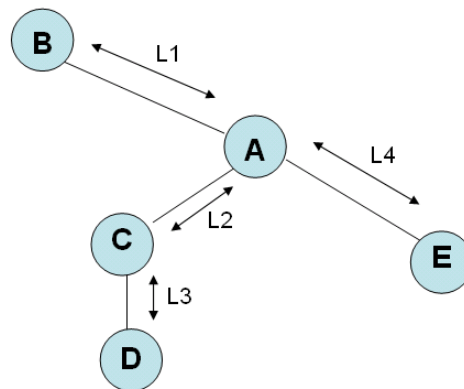


Abbildung 6: Raumgraph

weder Raumknoten A noch Raumknoten B bekannt. Raumknoten A sieht also nach dem Erhalt der Sichtungsnachricht von Lesegerät L1 in seinem lokalen Datenspeicher nach, ob ihm diese ID bekannt ist. Die ID ist A nicht bekannt. A sendet daraufhin die Sichtungsnachricht von L1 zusammen mit der Information, dass A die ID unbekannt war, an B. B sieht daraufhin in seinem lokalen Datenspeicher nach, ob ihm diese ID bekannt ist. In diesem Fall ist auch B die gesichtete ID unbekannt. B erkennt nun, dass ein *Slide* eingetreten ist. Ein *Slide* stellt ein Ereignis dar, dass eine Transponder-ID auf seinem Weg durch das Tracking-System von einem oder mehreren Lesegeräten nicht gesehen wurde - die ID also von einem Raum in einen anderen Raum unbemerkt „übergeschliddert“ ist⁹. Über diesen Slide informiert B nun den Tracking-Server durch die Nachricht von L1 zusammen mit der Information, dass die ID sowohl A wie auch B unbekannt war. Der Tracking-Server schaut nun nach, ob ihm und somit auch anderen Raumknoten in seinem Einflussbereich die ID bereits bekannt ist. Wenn die ID auch dem Tracking-Server unbekannt ist, meldet er diesen Slide seinem Vaterknoten-Tracking-Server in der übergeordneten Tracking-Architektur, die später in Kapitel 6 vorgestellt wird (im Sequenzdiagramm ist die Meldung an den Vaterknoten nicht dargestellt).

2.) Als gerade eben Raumknoten A und B, sowie der Tracking-Server festgestellt haben, dass sie die ID noch nicht kannten, haben sie die ID zusammen mit dem Sichtungszeitpunkt aus der Sichtungsnachricht als zusammengehörendes Tupel in ihren lokalen Datenspeicher übernommen. Die ID ist von nun ab also sowohl A und B, wie auch dem Tracking-Server bekannt. Nun erkennt L1 die selbe ID von gerade eben noch einmal. L1 sendet wieder eine

⁹Es ist ebenfalls möglich, dass das Objekt mit der betreffenden Transponder-ID einen Weg genommen hat, welcher in der CAD-Zeichnung nicht eingetragen ist - durch ein Fenster zum Beispiel. Desweiteren ist zu bedenken, dass eine ID auch überhaupt das allererste mal im gesamten Tracking-System aufgetaucht sein kann. So eine „Ersterscheinung“ kann lokal aber nicht von einem Slide unterschieden werden

Left by Slide Meldung an
Vater-Knoten-Tracking-Server

Nr	Msg	Verschärfung	LbS
1	A v B		
2	A v B	A	
3	A v C	C	
4	C v D		
5	C v D		
6	C v D		1
7	A v E		
...

Abbildung 7: Tracking-Tabelle

Sichtungs-Nachricht an Raumknoten A. Nun kennt Raumknoten A jedoch bereits die ID und leitet die Sichtung-Nachricht von L1 zusammen mit der Information, dass A die ID kennt an B weiter. B sieht auch, dass er die ID bereits kennt und stellt fest, dass sowohl A wie auch B diese Information bereits bekannt war. Es ist also keine neue Information die an den Tracking-Server weitergeleitet werden muss. Das einzige, was die beiden Raumknoten A und B tun müssen, ist den alten Sichtungszeitpunkt aus dem zuvor gespeicherten Tupel durch den neueren Sichtungszeitpunkt zu ersetzen.

3.) Nun erkennt L2 die besagte ID und schickt eine Sichtung-Nachricht an A. Der Raumknoten A schickt die Sichtung-Nachricht von L2 zusammen mit der Information, dass die ID dem Raumknoten A bereits bekannt ist an C weiter. C kennt die ID nicht und speichert darum nun ein ID-Zeit-Tupel in seinem lokalen Datenspeicher ab. Zudem leitet Raumknoten C die Sichtung-Nachricht weiter an den Tracking-Server zusammen mit der Information, dass zwar A die ID bereits kannte, C aber nicht. Anders als im Durchlauf eins kennt der Tracking-Server nun aber die ID bereits. Der Tracking-Server kann darum in seiner Datenbank nachsehen, welchem Raumknoten, der nicht Raumknoten A ist, die besagte ID bekannt ist. Diesem Raumknoten (in diesem Fall B) sendet der Tracking-Server sodann eine Lösch-Nachricht, welche besagt, dass der betreffende Raumknoten die ID aus seinem lokalen Datenspeicher löschen soll¹⁰. Der Tracking-Server weiss nun, dass die ID sich zwischen dem Sichtungszeitpunkt von L2 und dem Sichtungszeitpunkt der allerletzten Sichtung durch L1 in A aufgehalten haben muss. Leider wurde in Schritt zwei der allerletzte Sichtungszeitpunkt von L1

¹⁰Die Lösch-Nachricht beinhaltet den Sichtungszeitpunkt aus der auslösenden Sichtung-Nachricht. Sollte bei dem Raumknoten, welcher die Lösch-Nachricht erhält, bereits eine neuere Sichtungsnachricht zu der betreffenden ID eingetroffen sein, wird die Lösch-Nachricht ignoriert

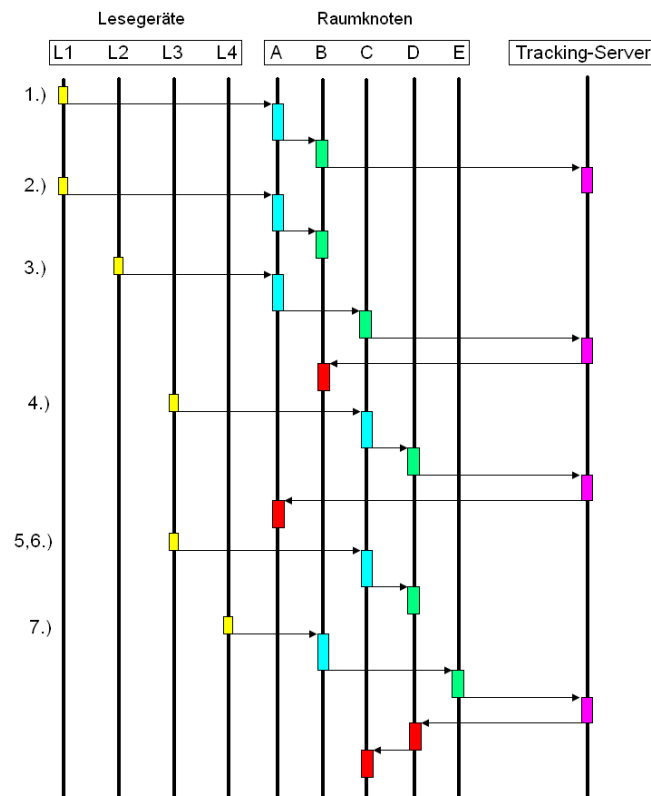


Abbildung 8: Sequenz-Diagramm

nicht an den Tracking-Server weitergeleitet, sodass in der L2-Sichtungs-Nachricht von C an den Tracking-Server zusätzlich noch der letzte Sichtungszeitpunkt, der in C gespeichert war, mitgeschickt werden muss (der letzte gespeicherte Sichtungszeitpunkt in C war der letzte L1 Sichtungszeitpunkt). Nur so kann man im Nachhinein feststellen in welchem Zeitraum der Aufenthalt der ID unscharf in Raum AvB war und in welchem Zeitraum scharf in A.

4,5,6.) In 4, 5 und 6 soll L3 mehrfach die besagte ID gesichtet haben. Diese Information geht analog zum vorher Beschriebenen an die beiden Raumknoten C und D - und zumindest zum Teil an den Tracking-Server.

7.) Nun wird angenommen, dass L4 die besagte ID sieht. Es erfolgt eine Sichtungsnachricht an A, welcher die ID nicht (mehr) kennt und durch A geht die Nachricht weiter an E, welcher die ID ebenfalls nicht kennt. E erkennt also wie unter Punkt 1 einen Slide. Anders als unter Punkt 1 bekommt nun der Tracking-Server die Slide-Nachricht und kennt jetzt aber die ID bereits. Analog zu Punkt 3 wird nun an beide bis hierhin als Aufenthaltsbereich angenommenen Raumknoten C und D eine Löschnachricht versendet. Hier zwar nicht dargestellt, aber dennoch sinnvoll wäre eine Meldung von C bzw. D zurück an den

Tracking-Server, in welcher stehen würde, wann C bzw. D die betreffende ID das letzte mal gemeldet bekommen haben.

8.) Im Sequenzdiagramm nicht dargestellt, ist das Ereignis, dass der Trackingserver von seinem Vaterknoten-Tracking-Server aus der übergeordneten Tracking-Architektur eine Lösch-Nachricht für eine bestimmte ID erhält (ausgelöst durch einen Slide in einem anderen lokalen Tracking-System). In diesem Fall müsste der Tracking-Server in seiner Datenbank nachsehen, welchen Raumknoten die ID aktuell bekannt ist und diesen Raumknoten eine Lösch-Nachricht schicken.¹¹

4 Fremde Datenquellen

Soeben wurde beschrieben, wie RFID-Lesegeräte durch RFID-Lese-Events Daten an das Tracking-System liefern. Ein einzelnes Lese-Event wie „A v B“ liefert nur eine unscharfe Position, die sich tatsächlich in Raum A oder B befinden könnte. Erst zwei aufeinander folgende und unterschiedliche Lese-Events können eine nachträgliche Verschärfung des Aufenthaltsortes hin zu zum Beispiel Raum A ergeben. Dieses kann man auch sehr schön in Mengenschreibweise darstellen:

$$\{A,B\} \cap \{A,C\} = \{A\}$$

Dabei ist $\{A,B\}$ ein konkretes Lese-Event und $\{A,C\}$ ein anderes.

Stellen wir uns nun vor, dass wir mehrere Fastfood-Filialen haben, die an ein und demselben Abrechnungssystem angeschlossen sind. Angenommen wir können aus dem Abrechnungssystem Informationen darüber erhalten, dass eine bestimmte Person gerade etwas eingekauft hat. Leider können wir aber nicht sehen, in welcher unserer Filialen der Einkauf getätigt wurde. Der unscharfe Bereich könnte so aus den Räumen B, F, H und L bestehen. Angenommen unsere letzte RFID-Sichtung liefert die möglichen Räume A und B, so erhalten wir:

$$\{A,B\} \cap \{B,F,H,L\} = \{B\}$$

Über solche „fremden Datenquellen“ könnte auch das Problem gelöst werden, dass für Räume, die „Blätter“¹² im Graphen darstellen, keine Verschärfung möglich ist, wenn ausschließlich das System aus Kapitel 3 zur Anwendung kommt.

¹¹Zu den Herausforderungen der Zeitsynchronisation, sowie der ID-gebundenen optimistischen Transaktionen zwischen Lesegeräten, Raumknoten und Trackingservern und auch der Fehlerkompensation bei verzögerter Nachrichtenübermittlung sowie bei ausgefallenen Knoten kann zum Bedauern des Autors an dieser Stelle aus Gründen des Umfangs leider nicht näher eingegangen werden. Das soll aber nicht heißen, dass diese Themen unwichtig wären.

¹²Da der Raum-Graph Zyklen aufweisen kann, ist es eigentlich allgemein gesprochen kein Baum, aber man sollte verstehen, was hier mit „Blättern“ gemeint ist. Korrekter wäre wohl die Aussage „Raumknoten mit nur einem adjazenten Raumknoten“

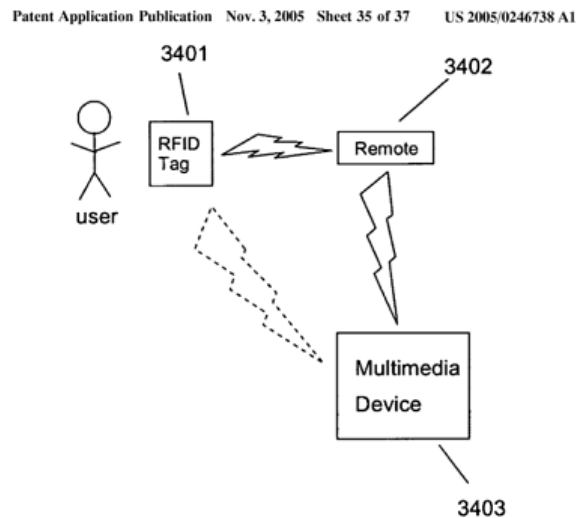


Abbildung 9: Auszug aus einem Patentantrag der Firma TiVo

Im Fall unseres Ferienclubs könnten wir uns zum Beispiel auch vorstellen, dass die Fernbedienung für die Steuerung der HiFi-Video-Anlage eines Raumes ein RFID-Lesegerät beinhaltet, welches den RFID-Transponder am Handgelenk des Gastes erkennen könnte (siehe dazu auch den US-Patentantrag 20050246738 der Firma TiVo, der so eine Fernbedienung vorsieht). Wenn die Fernbedienung ein Gerät aus dem Raum B ansteuert, könnte das Ergebnis für die dabei gelesene ID wie folgt aussehen:

$$\{A, B\} \cap \{B\} = \{B\}$$

5 Mobile Lesegeräte

Ähnlich könnten auch mobile RFID-Lesegeräte, die zum Beispiel von Club-Mitarbeitern oder Service-Robotern getragen werden, zu zusätzlichen Verschärfungen beitragen. Dazu müssten die mobilen RFID-Lesegeräte als erstes auf irgendeine Weise ihre aktuelle Position feststellen. Das könnte durch Laufzeitmessungen mit anschließender relativen und dann absoluten Positionsbestimmung geschehen, wie es bereits in Kapitel zwei mit dem Verweis auf UWB angedeutet wurde. Denkbar wäre an dieser Stelle aber auch ein RFID-Landmarkensystem, wie es unter [Stein 2005a] beschrieben ist¹³. Ein mobiles Lesegerät könnte dann zum Beispiel die „Teilraum“-Positionen {A1, A2} bzw. {A} angeben. Um Überschneidungen des Lesebereichs eines mobilen Lesegerätes mit mehreren Räumen gleich-

¹³Beim Landmarkensystem könnten ebenfalls nachträglich Verschärfung der Lesegerät-Position notwendig werden, sodass die darauf folgende Verschärfung der gesichteten ID nur nachträglich möglich wird

zeitig zu vermeiden, sollten Landmarken um stationäre RFID-Lesegeräte dafür sorgen, dass mobile RFID-Lesegeräte wissen, wann sie Objekt-ID Sichtungen besser ignorieren sollten, da sie an dem Ort nicht genau sagen können, in welchem Raum sich die ID nun tatsächlich befindet.

6 Übergeordnete Tracking-Architektur

In Kapitel drei wurde bei der Beschreibung der Vorgänge, die im Sequenzdiagramm in Abbildung 8 zu sehen sind, mehrmals auf die hier ebenfalls entwickelte übergeordnete Tracking-Architektur verwiesen. Im Punkt 1 steht dort „Wenn die ID auch dem Tracking-Server unbekannt ist, meldet er diesen Slide seinem Vaterknoten-Tracking-Server in der übergeordneten Tracking-Architektur [...]“ (siehe Abb. 10). Angenommen der *Tracking-Server 2* sei der Tracking-Server, der soeben festgestellt hat, dass diese ID neu in seinem Bereich ist. Er meldet also dem *Tracking-Server 1* einen Slide. Dieser wiederum sieht in seiner Datenbank nach, ob ihm die ID bekannt ist. Wenn dem *Tracking-Server 1* die ID bekannt ist, schickt er an den Teilbaum, in dem sich die ID zuletzt befunden hat, eine Lösch-Nachricht. In diesem Fall sei das der Kinds-knoten *Tracking-Server 3*. Der *Tracking-Server 3* handelt daraufhin, wie in Kapitel drei Punkt acht beschrieben wurde. Nicht zu vergessen ist natürlich, dass der *Tracking-Server 1* nun noch für die betreffende ID den aktuellen Aufenthaltsraum von Kinds-knoten 3 (*Tracking-Server 3*) auf Kinds-knoten 2 (*Tracking-Server 2*) ändern muss.

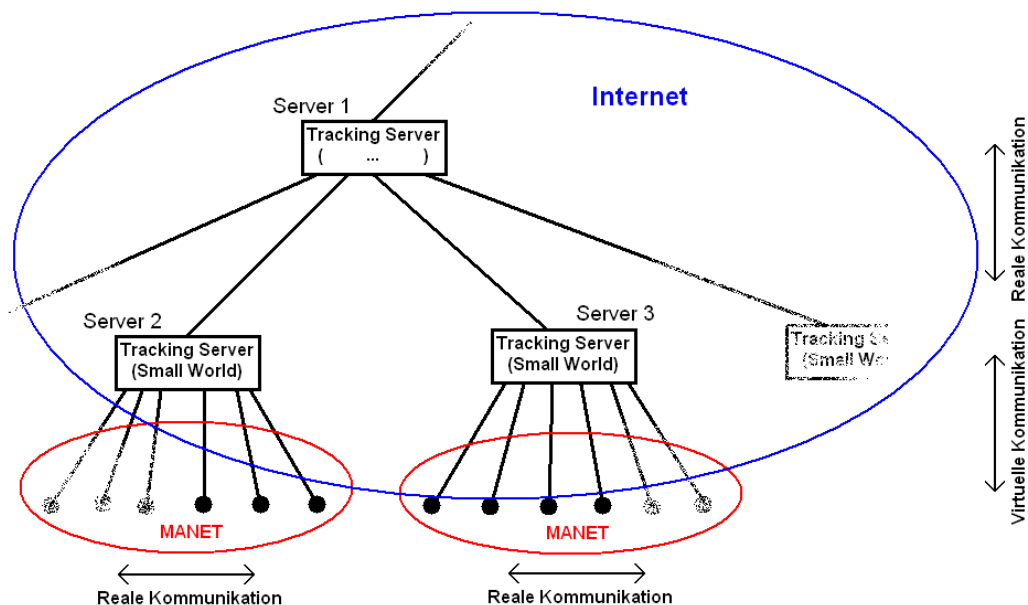


Abbildung 10: Tracking-Architektur inkl. „übergeordneter“ Tracking-Architektur

Wenn hingegen dem *Tracking-Server 1* die ID ebenfalls unbekannt gewesen wäre, hätte

er den Slide noch weiter nach oben an seinen Vaterknoten geleitet. Wenn es sich um eine ganz neue ID handeln sollte, wird der Slide bis zur Wurzel (dem Welt-Tracking-Server¹⁴) geleitet.

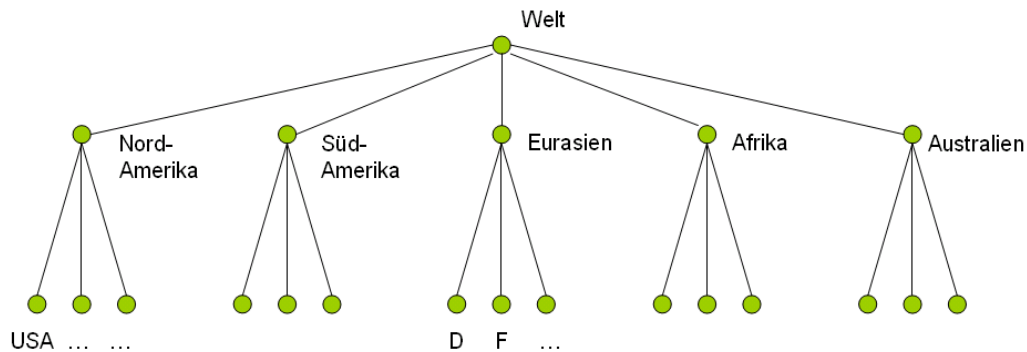


Abbildung 11: Weltsystem

Die einzelnen Tracking-Server speichern dabei ausschließlich, in welchem Teilbaum unter ihnen sich die ID befindet. Sollte man also eine Positionsanfrage zu einer ID an das System stellen wollen, würde man seinen lokalen Tracking-Server befragen. Wenn der lokale Tracking-Server die Information nicht hat, fragt er seinen Vaterknoten. Diese Anfrage geht so lange weiter im Baum, bis ein Vaterknoten gefunden wurde, der die ID kennt. Daraufhin wird der Baum nach unten entlang der Kindsknoten traversiert, welche die ID kennen, bis man am untersten Tracking-Server angekommen ist, welcher einem die genaue Position verraten kann. Wenn nur eine allgemeinere Positionsanfrage gestellt wird¹⁵, kann bei diesem System selbstverständlich auch früher abgebrochen werden.

7 Wegvorhersage

Ein Tracking-System bietet einem die Möglichkeit die aktuelle Position eines Objektes in Erfahrung zu bringen. Zudem können wir aber auch viel über die Bewegungshistorie eines Objektes herausfinden. Auf Basis dieser historischen Daten können wir dann zudem versuchen, die statistischen Wahrscheinlichkeiten für den zukünftigen Fortgang der angefangenen Bewegung zu berechnen.

Eine Möglichkeit wäre es, wie folgt vorzugehen. Nehmen wir an, wir hätten einen Raumknoten A, welcher direkte Übergänge zu den Knoten B, C und D besitzt. Nun würden wir

¹⁴Es sei hier nur am Rande angemerkt, dass die einzelnen Tracking-Server bis hinauf zum Welt-Tracking-Server, zur Skalierung selbstverständlich auch durch mehrere Server repräsentiert werden können, wobei zum Beispiel nur bestimmte ID-Bereiche von einem Teil-Server verwaltet werden

¹⁵Wenn es zum Beispiel reicht zu wissen, ob sich die ID gerade in einer bestimmten Stadt aufhält

bei jedem Übergang von Raum A zu einem der anderen drei Räume einen Zähler an der verbindenden Kante um eins hoch zählen. Um die unmittelbare Übergangswahrscheinlichkeit von Knoten A zu z.B. Knoten B zu berechnen, müssten wir nur den Zähler an der Kante von A nach B durch die Summe der Kanten-Zähler, welche zu A inzident sind und von A weggehen, teilen. Auf Basis dieser Übergangswahrscheinlichkeiten könnte man nun mit Hilfe von Markow-Ketten versuchen Aussagen über die mögliche Zukunft zu machen. So könnte man zum Beispiel berechnen, wie wahrscheinlich es ist, dass das Objekt nach vier Schritten auf Knoten K ist (egal über welchen Weg). Auch Aussagen über den am wahrscheinlichsten gleich beschrittenen Weg sind möglich. Aber diese Berechnungen zielen nur auf das Verhalten der einzelnen Objekte ab. Besonders interessant wird es hingegen, wenn der Zustandsraum nicht nur ein einzelnes Objekt vorsieht, sondern viele Objekte. So könnte man auf Basis der einfachen Übergangswahrscheinlichkeiten und der anschließenden Betrachtung auf Basis von Markow-Ketten die Wahrscheinlichkeit berechnen, dass es nach x Schritten zu einem „Stau“ vor dem Frühstücksbuffet kommt.

Den Autor störte hingegen, dass Markow-Ketten durch die Markow-Eigenschaft ohne Gedächtnis sind und somit ein vorheriges Bewegungsverhalten keinen Einfluss auf die Wahrscheinlichkeiten der nächsten Schritte hat. Ein Beispiel: Stellen wir uns einen Flur vor, an den vier Räume angrenzen - ein Fernsehzimmer, ein Esszimmer, ein Raum mit Swimmingpool und eine Duschaum mit Umkleidekabine. Stellen wir uns nun vor, dass eine Person aus dem Raum mit Swimmingpool auf den Flur austritt und dass eine zweite Person aus dem Fernsehzimmer ebenfalls auf den Flur austritt. Nach Markow wären die Wahrscheinlichkeiten hiernach weiter in das Esszimmer zu gehen für beide Personen gleich. Ob das tatsächlich der Fall ist, darf man wohl bezweifeln - zumindest wenn es nach den Landessitten und Gebräuchen unüblich ist, klatschnass und in Badehose zu dinieren.

Die Lösungsalternative einer Wegvorhersage sah darum wie folgt aus. In einer Datenbank im lokalen Tracking-Server sollte das tatsächliche Beschreiten eines Wegstücks gezählt werden. Ein Tupel $(n_1, n_2, n_3, n_4, n_5, c, t)$ ¹⁶ bestand aus fünf Knoten-IDs, einem Zahlenwert (Zähler) und einem Zeitstempel (letztes auftreten dieses Wegstücks). Zu einer ID wurden temporär immer die letzten vier beschrittenen Knoten-IDs gespeichert. Wenn eine neue Knoten-ID betreten wurde, wurde ein Eintrag in die Datenbank gesucht, wobei die letzten vier beschrittenen Knoten-IDs mit n_1 bis n_4 und n_5 mit dem neuen Knoten übereinstimmen musste. Wenn so ein Eintrag vorhanden war, wurde c um eins inkrementiert und t auf die aktuelle Zeit gesetzt. Wenn so eine Zeile nicht vorhanden war, wurde sie neu angelegt und c mit eins initialisiert.

Wenn man nun die Wahrscheinlichkeiten für die nächsten zwei Schritte berechnen wollte, nahm man die letzten drei Schritte aus dem temporären Speicher und suchte in der Relation die Tupel in der n_1 bis n_3 mit den letzten drei Schritten übereinstimmte. Beispiel:

¹⁶Tatsächlich war es $(n_1, n_2, n_3, n_4, n_5, \text{type}, c, t)$, wobei type einen Objekttypen darstellte, der Gast, Mitarbeiter oder Inventar sein konnte

(A,B,E,F,K,4,t1), (A,B,E,F,I,7,t2), (A,B,E,B,K,9,t3). Die Wahrscheinlichkeit für das erste Tupel ist $4/(4+7+9)$, für das zweite Tupel $7/(4+7+9)$ und für das dritte Tupel $9/(4+7+9)$. Also mit einer Wahrscheinlichkeit von $4/(4+7+9) + 7/(4+7+9)$ wird der nächste Schritt F sein und mit einer Wahrscheinlichkeit von $4/(4+7+9)$ wird der nächste Weg F,K sein¹⁷, aber die Wahrscheinlichkeit für das Erreichen von K im übernächsten Schritt (ohne Berücksichtigung des Weges) ist wieder $4/(4+7+9) + 9/(4+7+9)$.

Über dieses System war dann also eine einfache Wegvorhersage mit Einbeziehung von Vorbedingungen möglich. Wenn man weiter in die Zukunft schauen möchte, muss man weitere Knoten-Spalten einführen. Wie sehr dabei der Speicheraufwand in der Datenbank ansteigt, hängt zum einen sehr stark davon ab, wie der Graph aufgebaut ist, in der sich die Objekte bewegen und zum anderen davon, wie chaotisch sich die Objekte im Graph bewegen. Um den Speicheraufwand über die Zeit zu zügeln, kann man Tupel mit kleinen Zählerwerten und alten Zeitstempeln löschen. Tupel mit großen Zählerwerten und alten Zeitstempeln könnte man altern lassen, indem man abhängig vom letzten Auffrischen den Zählerwert inkrementiert.

Als Vereinfachung der Vorhersage von „Staus“ vor dem Frühstücksbuffet wurden mehrere Verfahren durchdacht, die alle aber nicht wirklich zufriedenstellend waren (zumindest in der Theorie). Eine Vereinfachung sieht zum Beispiel vor, dass nur die wahrscheinlichsten zukünftigen Wege der einzelnen Objekte angesehen werden, um zu überprüfen, wie viele dieser Wege sich bald am Buffet kreuzen.

Markow-Ketten scheinen also besser geeignet zu sein, um weiter in die Zukunft zu schauen und um Wahrscheinlichkeiten für Gruppenverhalten zu berechnen. Dafür hat dieses kleine Vorhersageverfahren mit Wegteilstücken den Vorteil, dass es auch Vorbedingungen mit in die Wahrscheinlichkeiten mit einbezieht.

Um vielleicht einen Weg zu finden, diese beiden Welt zu vereinen, wurde dann noch kurz versucht dieses Wegteilstück-Verfahren auf Markow-Ketten zu mappen.

Ein Beispiel sehen wir in Abbildung 12.

Wir erhalten aus den Tabelle also kleine „Markow-Bäume“ (siehe dazu auch „Darstellungsmöglichkeiten - Baumdiagramm“¹⁸). Zu beachten ist, dass die Vorbedingungen in die Zustände trickreicherweise „einkodiert“ sind. Für die korrekte Baumdarstellung müssen wir die Bäume noch, wie in Abbildung 13 dargestellt, „aufbrechen“.

Mit diesen direkten Mappings können wir wohl relativ wenig anstellen. Basierend auf der Idee von „Markow-Bäumen“ mit Vorbedingungsgeächtnis in den Zuständen, können wir nun aber versuchen, die Wahrscheinlichkeiten für noch längere Wege in der Zukunft zu berechnen. Hierzu verketteten wir die bestehenden Wegblöcke aus der Datenbank über ihre jeweilig überlappenden Übergangswahrscheinlichkeiten - siehe Abbildung 14.

¹⁷Wegwahrscheinlichkeiten könnten für Wegkreuzungswahrscheinlichkeiten bei mehreren bewegten Objekten interessant werden

¹⁸<http://www.siegel-christian.de/seiten/facharbeit/markow.html#in241>

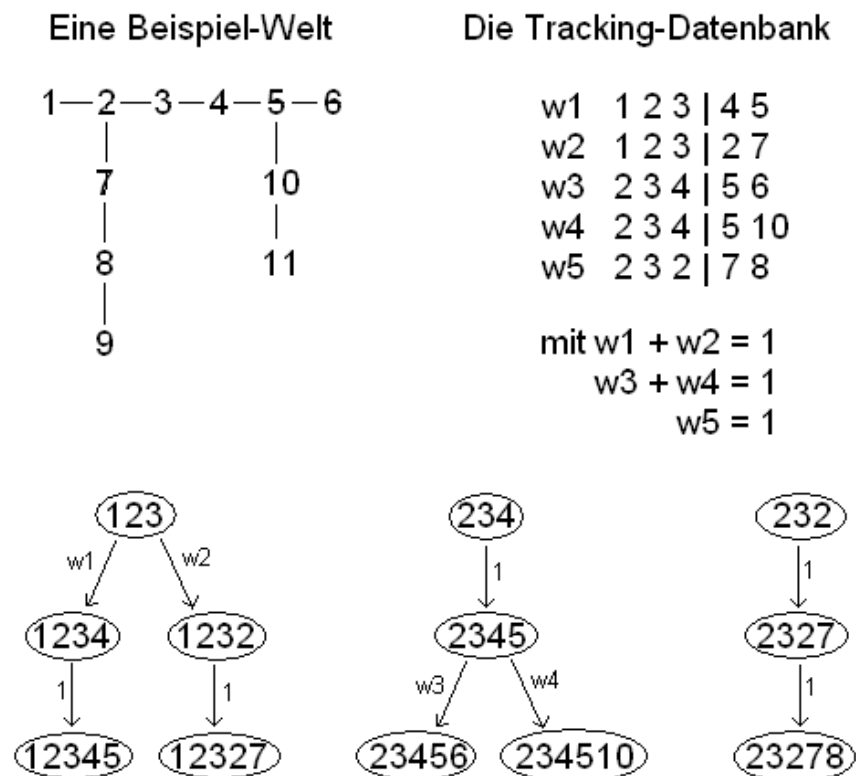


Abbildung 12: Mapping Wegstück-Datenbank zu Markow-„Bäumen“

Diese Markow-Ketten Darstellung als Baumdiagramm können wir leicht in die Darstellungsform als Kreisdiagramm bzw. Zustandsgraphen überführen, indem wir alle Teilstück (das Beispiel in der Abbildung zeigt nur ein paar Teilstücke aus der Datenbank) richtig miteinander verbinden (siehe hierzu „Darstellungsmöglichkeiten - Kreisdiagramm“¹⁹).

Nun könnten wir wie in den ursprünglichen Markow-Ketten einen Zustandsraum definieren, in dem sich mehrere Objekte bewegen. Nun könnten wir wiederum Wahrscheinlichkeiten von „Staus“ auf einem Knoten suchen. Nun stellt ein Knoten jedoch anders als zuvor nicht mehr einen einzelnen Raum dar, sondern ein Wegstück. Uns interessiert aber nicht die geringere „Gefahr“, dass viele Objekte ein Wegstück bestehend aus mehreren Räumen gemeinsam begehen, sondern dass sie gemeinsam in einem einzelnen Raum sind. Um diese Wahrscheinlichkeit zu ermitteln, müssten wiederum mehrere Wegstückwahrscheinlichkeiten miteinander addiert werden. Dieser Ansatz konnte jedoch aus Zeitgründen bisher leider nicht weiter verfolgt werden. In wieweit sich der Rechenaufwand durch eine geschickt gewählte

¹⁹<http://www.siegel-christian.de/seiten/facharbeit/markow.html#in242>

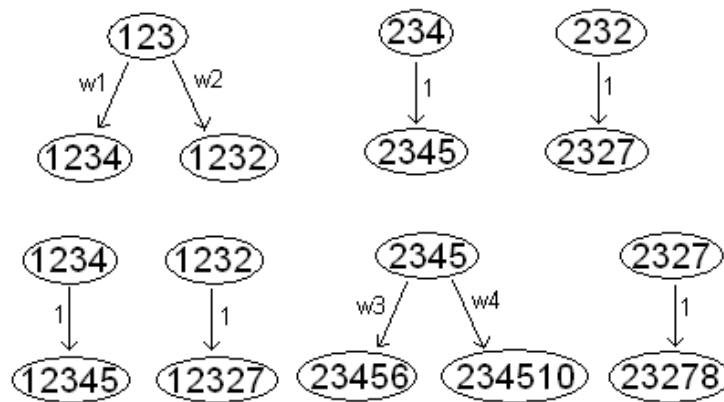


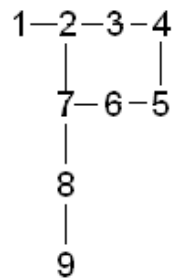
Abbildung 13: Markow-Ketten in Baumdarstellung

Datenbankstruktur und ein lokales Daten-Update bei Bewegungen innerhalb des Graphen optimieren ließe, wurde an dieser Stelle ebenfalls noch nicht genauer untersucht.

8 Versuchs-Implementation

Um die theoretischen Überlegungen zumindest in Teilen zu überprüfen, wurde eine Versuchs-Implementation realisiert. Zudem hat diese Realisierung zur Weiterentwicklung der Ideen beigetragen, da aufkommende und zuvor nicht bedachte Probleme gelöst werden mussten. Die Versuchs-Implementation ist in Java 2 Standard Edition 1.4.2 geschrieben. Alle Knoten des Trackingsystems (RFID-Lesegeräte, virtuelle Raumknoten und Tracking-Server) sind als einzelne Threads realisiert. Die Threads kommunizieren miteinander, indem sie nach der Instanziierung bei einem Objekt der Klasse TrackerMiddleware angemeldet werden. Die Klasse TrackerMiddleware abstrahiert von der Erfordernis eines Nachrichtenroutings innerhalb des Netzwerkes und sorgt dafür, dass jeder Knoten an jeden anderen Knoten ein Datenpaket senden kann, indem er der TrackerMiddleware das Datenpaket übergibt. Anhand des Destination-Feldes eines solchen Datenpakets, weiß die TrackerMiddleware, an welchen der Knoten sie das Datenpaket übergeben muss. Zur Datenpersistenz innerhalb des Tracking-Servers wird eine MySQL 4.1 Datenbank verwendet. Da auf exotische Datentypen verzichtet wurde und der Zugriff auf die Datenbank per JDBC erfolgt (MySQL Connector/J 3.1.1.12), kann die Datenbank auch relativ leicht durch andere Datenbanken wie zum Beispiel den Microsoft SQL-Server ausgetauscht werden. Da mit der Implementation dieses Versuchs-Systems bereits begonnen wurde, bevor überhaupt das Gesamtkonzept relativ vollständig

Eine Beispiel-Welt



Die Tracking-Datenbank

w1	1 2 3	4 5
w2	1 2 3	2 7
w3
w4	2 3 4	5 6
w5	2 3 4	5 4
w6	2 3 2	7 8
w7	2 3 2	7 6
w8
w9	3 4 5	6 7

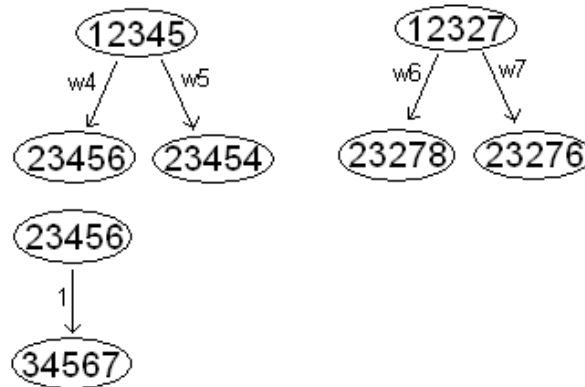


Abbildung 14: Markow-Ketten mit „versteckten“ Vorbedingungen

ausgearbeitet war, und somit an manchen Stellen eher „geflickt“ als entwickelt wurde, sind einige Komponenten für Erweiterungen einfach zu unflexibel aufgebaut. Zur Verdeutlichung des hier entwickelten Konzeptes sollte es aber problemlos reichen. Nur von einer Weiterentwicklung basierend auf der bestehenden Code-Basis sollte wohl besser verzichtet werden.

Literatur

- [BITKOM 2005a] BITKOM: *White Paper - RFID - Technologie, Systeme und Anwendungen*. 2005. – URL http://www.bitkom.org/files/documents/White_Paper_RFID_deutsch_11.08.2005__final.pdf
- [BITKOM 2005b] BITKOM: *White Paper - RFID - Technology, Systems, and Applications*. 2005. – URL http://www.bitkom.org/files/documents/White_Paper_RFID_english_12.12.2005_final.pdf
- [Feng Zhao 2004] FENG ZHAO, Leonidas J. G.: *Wireless sensor networks - an information processing approach*, ISBN 1-558-60914-8. 2004
- [Flörkemeier 2005] FLÖRKEMEIER, Christian: EPC-Technologie - vom Auto-ID Center zu EPCglobal. In: *Das Internet der Dinge*, ISBN 3-540-24003-9, 2005, S. 98
- [Germany 2005] GERMANY, GS1: *Internet der Dinge - Management-Information*. 2005. – URL http://www.gs1-germany.de/internet/common/daten/epcglobal/files/mip/internet_der_dinge.pdf
- [Heinrich 2005] HEINRICH, Claus: *RFID and Beyond - Growing Your Business Through Real World Awareness*, ISBN 0-7645-8335-2. 2005
- [RadioActive 2005] RADIOACTIVE: *RadioActive Neutrino, Fusion und Graviton*. 2005. – URL <http://www.radioactivehq.org>
- [Refactored-Networks 2005] REFACTORED-NETWORKS: *xcEPC (eXtreme Commerce EPC) Network Core for Java*. 2005. – URL <http://www.refactored-networks.com/downloads.html>
- [Stein 2005a] STEIN, Martin: *Entwicklung eines auf RFID basierenden mobilen Objekt-Tracking-Systems*. 2005. – URL <http://users.informatik.haw-hamburg.de/~ubicomp/arbeiten/diplom/stein.pdf>
- [Stein 2005b] STEIN, Martin: *Mit RFID-Technologie zum Internet der Dinge*. 2005. – URL <http://users.informatik.haw-hamburg.de/~ubicomp/projekte/master05-06/stein/abstract.pdf>
- [Stein 2006] STEIN, Martin: *Mobile Ad-hoc Networks und Wireless-Sensor-Networks*. 2006