Diplomarbeit

Martin Stein

Entwicklung eines auf RFID basierenden mobilen Objekt-Tracking-Systems

Fachbereich Elektrotechnik und Informatik
Department of Electrical Engineering and Computer Science

Martin Stein

Entwicklung eines auf RFID basierenden mobilen Objekt-Tracking-Systems

Diplomarbeit eingereicht im Rahmen der Diplomprüfung
im Studiengang Technische Informatik
am Fachbereich Elektrotechnik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

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

Abgegeben am 21. Februar 2005

Martin Stein

Thema der Diplomarbeit

Entwicklung eines auf RFID basierenden mobilen Objekt-Tracking-Systems

Stichworte

RFID, Radio Frequency Identification, Tracking, Positionsbestimmung, Ubiquitous Computing, PDA, µController, Bluetooth, PersonalJava

Kurzzusammenfassung

Diese Diplomarbeit beschäftigt sich mit der Funktechnologie „RFID“. In diesem speziellen Fall geht es um eine Objektverfolgungs- und Überwachungsanwendung durch ein RFID-Lesegerät mit angeschlossenem µController. Vom µController gesammelte Informationen können im Bedarfsfall durch einen PDA über Bluetooth abgefragt werden. Über den PDA ist der Anwender in der Lage Überwachungsregeln aufzustellen, die der µController selbständig zu kontrollieren hat.

Martin Stein

Title of the paper

Development of an Object-Tracking-System based on RFID

Keywords

RFID, Radio Frequency Identification, Tracking, position sensing, Ubiquitous Computing, PDA, µController, Bluetooth, PersonalJava

Abstract

This thesis deals with the radio technology „RFID“. This case handles especially an application of object tracking and monitoring by an RFID-Reader with a connected µController. Information collected by the µController can be queried by a PDA over a Bluetooth connection. The user is able to establish monitoring rules by using a PDA. Afterwards the µController checks the compliance with regulations autonomously.

Danksagung

Hiermit möchte ich mich bei meinen Professoren, Kommilitonen und Freunden, die mich sowohl im Studium wie auch bei dieser Diplomarbeit fachlich als auch moralisch unterstützt haben, herzliche bedanken. Ein ganz besonderer Dank gilt jedoch meinen Eltern, durch die mein Studium und somit auch diese Diplomarbeit überhaupt erst möglich wurde. Danke!

Inhaltsverzeichnis

Kapitel 1

Einleitung

RFID[1] ist plötzlich in aller Munde. Die Öffentlichkeit oder vielmehr ihr Sprachrohr und bisweilen auch Vordenker die Tages- und Boulevardpresse befürchtet den gläsernen Menschen, der keinen Schritt mehr machen kann, ohne dass es Staatsmacht und Industriekonzerne registrieren würden. Im Gegensatz dazu berichten die Fachblätter der Wirtschafts- und Technologiepresse von wahren Wunderdingen, wenn nicht gar von einer utopischen Zukunft, in der zum Beispiel immense Kosten für Kassenpersonal eingespart werden kann, indem die Kunden von nun an beim Verlassen der Geschäfte quasi im Vorbeigehen automatisch bezahlen. Ladendiebstahl soll nur noch eine böse Erinnerung an längst vergangene Zeiten sein. Und die Logistik soll so perfektioniert sein, dass die Regale immer voll sein werden, ohne dass man weiterhin Kosten für eine aufwendige Lagerhaltung aufbringen müsste. Vieles davon wird wohl wahrscheinlich frommer Wunsch beziehungsweise nicht eingetretene Angstvorstellung bleiben, aber einiges wird ebenso gewiss seinen Weg in unser tägliches Leben finden. Ob die RFID-Technik mehr zu unserem Nutzen oder vielleicht doch eher zu unserem Schaden sein wird, kann wohl nur die Zukunft zeigen...

1.1 Motivation und Zielsetzung

Tatsächlich werden RFID-Systeme schon heute vermehrt in der Logistik eingesetzt, wo auch von führenden Wirtschaftsexperten das größte Wachstumspotential für RFID-Systeme in der näheren Zukunft erwartet wird. Seien es große Einzelhandelskonzerne wie die Metro

1 Radio Frequency Identification (englischsprachliche Bezeichnung für Funkerkennung)

Handelsgruppe [1], Automobilbauer wie DaimlerChrysler [2], Paketdienste wie DHL [3], Fluggesellschaften wie Delta Airlines [4] oder auch das amerikanische Militär [5], überall geht es im Grunde darum beziehungsweise soll es in Zukunft darum gehen, mit RFID-Systemen dafür zu sorgen, dass man von bestimmten Gegenständen immer genau weiß, zu welchem genauen Zeitpunkt sie bestimmte Ortspunkte auf ihrem Lebensweg passiert haben. Aus diesen gewonnenen Informationen soll dann abgeschätzt werden können, wo sich die einzelnen Gegenstände zurzeit befinden, sodass man dann der aktuellen Gesamtsituation entsprechend die Gegenstände zielgerichtet zur rechten Zeit an den rechten Ort verbringen kann. Im Grunde handelt es sich bei all den zur Zeit von der Industrie vorangetriebenen Projekten um verschiedene Ausprägungen von Objekt-Tracking-Systemen, also um Systeme, die dafür gedacht sind, Objekte in ihrer Ortslage zu verfolgen, um aus diesen gewonnenen Daten zu bestimmen, wie der weitere Lebensweg dieser und anderer Objekte verlaufen soll. Hierbei werden die zu verfolgenden Gegenstände mit RFID-Transpondern ausgestattet. An festgelegten Wegpunkten, an denen das Passieren von diesen RFID-Transpondern überwacht werden soll, werden jeweils RFID-Lesegeräte aufgestellt.

Bisher wird im Bereich des RFID-Objekt-Trackings primär für den Einsatz im industriellen Umfeld geforscht und entwickelt. Ein möglicher Einsatz und die direkte Nutzbarmachung für den privaten Konsumenten scheint bisher eher unbeachtet geblieben zu sein. Diese Marktlücke sollte mir dann auch der Anlass für das Thema dieser Diplomarbeit sein, in der es darum gehen soll, dem einzelnen Menschen ein komplettes Objekt-Tracking-System auf RFID-basis an die Hand zu geben. Die Aufgabe des Tracking-Systems soll dabei sein, den Benutzer darin zu unterstützen, Gegenstände seines täglichen Lebens wieder zu finden, beziehungsweise erst gar nicht zu verlieren.

1.2 Beschreibung der Aufgabe

Anders als bei den Systemen der Industrie soll hier nicht eine Vielzahl von ortsfesten Lesegeräten das Vorbeikommen einzelner mit RFID-Transpondern markierter Gegenstände protokollieren. In diesem hier zu entwickelnden System wird ein ortsveränderliches Lesegerät diesen Part übernehmen. Dieser Unterschied macht es dann auch notwendig, dass das Lesegerät nun zusätzlich seine eigene Position im Raum mehr oder weniger präzise auf irgendeine Weise abschätzen können muss, um verwertbare Protokolldaten zu erhalten.

Nun aber zu den genaueren Aufgaben, die das zu entwickelnde Tracking-System erfüllen soll. Der Benutzer soll gewarnt werden, wenn er zum Beispiel in Begriff ist, beim Verlassen des Büros seinen mit einem RFID-Tag markierten PDA liegen zu lassen. Auch soll darauf hingewiesen werden, wenn die ebenfalls mit einem RFID-Tag versehene Brieftasche oder auch der Aktenkoffer während der Fahrt in der U-Bahn plötzlich „seinen Besitzer wechselt“

oder nur aus Vergesslichkeit liegen bleibt. Aber auch bei der Suche nach schon verschwundenen Gegenständen soll das System helfen können, indem es einem verrät, zu welcher Uhrzeit man zuletzt in seiner Nähe war und welche Orte (RFID-Landmarken) man kurz zuvor und kurz danach passiert hat.

Dazu soll das Tracking-System inklusive des RFID-Lesegeräts durchgängig am Körper des Benutzers getragen werden. Es muss also klein und portabel sein, sowie möglichst lange unabhängig von externen Energiequellen operieren können. Eine Aufgabe des Tracking-Systems wird sein, die Umgebung des Benutzers fortwährend nach RFID-Tags abzusuchen. Wenn während der fortwährenden Suche nach RFID-Tags gegen eine Regel wie zum Beispiel „die mit einem RFID-Tag markierte Brieftasche soll immer in meiner Nähe sein“ verstoßen wird, soll das Trackingsystem den Benutzer akustisch alarmieren. Eine andere Aufgabe, die von dem System im Bedarfsfall bereitgestellt werden muss, ist es, dem Benutzer eine textbasierte oder graphische Menüführung zu bieten. Diese Menüführung soll es dem Benutzer ermöglichen, nähere Informationen über eine verletzte Regel einzuholen, neue Regeln anzulegen und bestehende Regeln zu bearbeiten. Zudem soll es dem Benutzer möglich sein, zu „trackende“ Objekte bzw. die RFID-Tags, die an ihnen angebracht sind, zu verwalten und im Bedarfsfall weitere Informationen über ihren Verbleib einzuholen.

Um diese beiden Aufgaben zu erfüllen, die zum einen aus einer fortwährenden Überwachung und zum anderen aus einer nur vereinzelt notwendigen Menüführung bestehen, bietet es sich an, diese beiden Aufgaben im Sinne eines verteilten Systems auch in Hardware voneinander zu trennen. Die Überwachung soll ein embedded System übernehmen, welches auf den Namen „RFID-Watch-Box“ getauft wird. Bestehen soll es aus einem RFID-Lesegerät und einem µController, sowie aus einer eigenständigen Energieversorgung. Für die Menüführung wird ein PDA vorgesehen, der in diesem Zusammenhang „RFID-Configurator“ heißen soll. Miteinander sollen diese beiden Systemkomponenten durch eine Bluetooth-Kommunikationsschnittstelle verbunden werden.

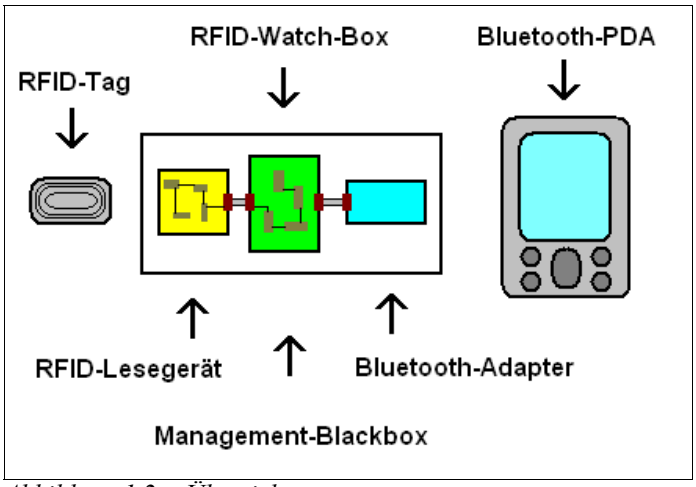

Abbildung 1.2a: Übersicht

Kapitel 2

Grundlagen

2.1 Beispiele aus aktuellen RFID-Anwendungen

Elektronische Artikelsicherungssysteme (EAS)

Viele der heute noch im Einsatz befindlichen elektronischen Artikelsicherungssysteme könnte man als frühe Vorfahren der modernen RFIDs ansehen. Ihre 1-bit-Transponder kennen aber nur zwei logische Zustände, welche im Rahmen der überwiegend anzutreffenden Anwendung mit den Worten „ich bin noch nicht bezahlt“ und „ich bin bereits bezahlt“ eindeutig beschrieben wären. Darüber hinaus tragen diese Transponder keine zusätzlichen Informationen. Viele von ihnen sind sogar nur ein einziges Mal in der Lage vom Zustand „ich bin noch nicht bezahlt“ nach „ich bin bereits bezahlt“ zu wechseln, da je nach Artikelsicherungssystem Teile des Transponders dafür irreversibel zerstört werden [6]. Im Grunde basieren sie alle auf einfachen physikalischen Effekten, wie zum Beispiel der Magnetostriktion von ferromagnetischen Metallen [7] und haben nur wenig mit den heutigen RFID-Tags gemeinsam.

Zugangskontrollsysteme

Eine gewisse Beliebtheit haben RFID-Systeme auch heute schon in Zugangskontrollsystemen gefunden. Ein großes Problem dieser Systeme war immer, dass die Benutzer zur Authentifizierung (Überprüfung der Identität) und Autorisierung (Einräumen bestimmter Rechte) eine zeitraubende Prozedur über sich ergehen lassen mussten. Normale Schließsysteme mit ihren Schlüssellöchern und Schlüsseln sind zwar mechanisch robust und

langlebig, die Benutzung ist aber relativ zeitintensiv. Zudem ist das Vertrauen in die Sicherheit solcher Systeme schwer erschüttert, seit es im Internet frei erhältliche, zerstörungsfreie Öffnungswerkzeuge wie so genannte „Elektro-Picks“ und „Schlagschlüssel-Sets“ gibt [8]. Zudem ist hier die Verwaltung der Zugangsfreigabe sehr kompliziert und unflexibel. Einen Fortschritt stellten elektronische Systeme auf Basis von Smartcards dar. Diese sind gegenüber den klassischen Schließsystemen um einiges flexibler in der Verwaltung und nehmen zumeist im täglichen Umgang mit den Benutzern weniger Zeit in Anspruch. Jedoch kommt es durch die ständigen Kontaktiervorgänge leicht zu Verschleißerscheinungen. Zudem ist die Vandalismusresistenz der Kartenschächte nicht höher als bei den herkömmlichen Schlüssellöchern. Hingegen ist die Vandalismusresistenz von RFID-Systemen durch die (meist) fehlende Einsteckeinheit um einiges erhöht. Durch das Ausbleiben von Kontaktiervorgängen ist zudem der Verschleiß stark reduziert und die normale Abfertigungszeit konnte durch den fehlenden Kontaktiervorgang verkürzt werden. In Systemen, bei denen die Schlüsselkarte noch nicht einmal mehr aus der Brieftasche geholt werden muss, ist die Verkürzung der Abfertigungszeit sogar beträchtlich.

Tieridentifikation

Die ISO-Normen 11784 und 11785 stellen ein einheitliches Verfahren zur Tieridentifikation bereit. Waren die ursprünglichen Bestrebungen in der Masttierhaltung noch darauf ausgerichtet, mit RFID-Systemen eine automatische Futterzuteilung und eine präzise Leistungserfassung zu realisieren, so hat es spätestens nach den gehäuften BSE-Fällen in der Vergangenheit und den Befürchtungen um die Ansteckungsgefahr für den Menschen, einen Zusatznutzen in der Seuchen- und Qualitätskontrolle gefunden. Die Anbringung der Transponder an die Tiere findet auf verschiedene Weise statt. Es gibt sowohl Halsband- und Ohrmarkentransponder, injizierbare Transponder, wie auch die so genannten Boli. Die injizierbaren Transponder sind kleine, längliche Glasstäbchen, welche mit Hilfe einer Hohlnadel unter die Haut des Tieres gebracht werden. Bei den Boli handelt es sich um säurebeständige Transponder, welche ausschließlich bei Wiederkäuern eingesetzt werden. Dabei wird über eine Sonde durch den Schlund des Tieres der Bolus in den Vormagentrackt (Pansen) verbracht. Unter normalen Bedingungen verbleibt er dort über die gesamte Lebensdauer des Tieres.

Abbildung 2.1a: Ohrmarken-Transponder

2.2 Technische Grundlagen

Die Grundidee hinter dem RFID-System ist es, Siliziumchips zur Speicherung von Daten zu haben, welche ihre Daten kontaktlos zu einem Lesegerät übertragen können. Zudem sollen diese Datenspeicher möglichst ohne eigene Energiequellen wie Batterien oder Netzteile auskommen. Stattdessen soll die Energie, die zum Betrieb der elektronischen Datenträger benötigt wird, ebenfalls kontaktlos durch das jeweils zugreifende Lesegerät bereitgestellt werden.

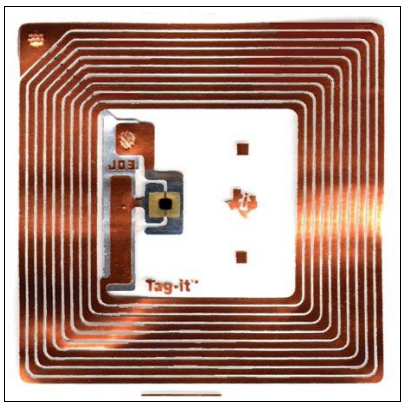

Abbildung 2.2a: Tag-it-Serie


Abbildung 2.2b: Mehrere passive RFID-Tags

RFID-Systeme werden bezüglich der Bauform der Transponder, der Daten- und Energieübertragung, der Übertragungsfrequenz, der Übertragungsrate, der Modulationsverfahren und der Reichweite unterschieden. Die zum Einsatz kommenden Transponder, auch „Tags“ genannt, werden grundsätzlich in die zwei Klassen der passiven und aktiven Transponder unterteilt. Passive Transponder beziehen ihre gesamte Energie aus dem Feld, welches das Lesegerät erzeugt, wohingegen die aktiven Transponder durch das Feld des Lesegeräts nur aus einem energiesparenden Tiefschlaf geweckt werden, um dann anschließend mit Hilfe einer Batterie mit dem Lesegerät zu kommunizieren. Nach dem Ende der Kommunikation fallen die aktiven Transponder wieder in ihren Tiefschlaf zurück.

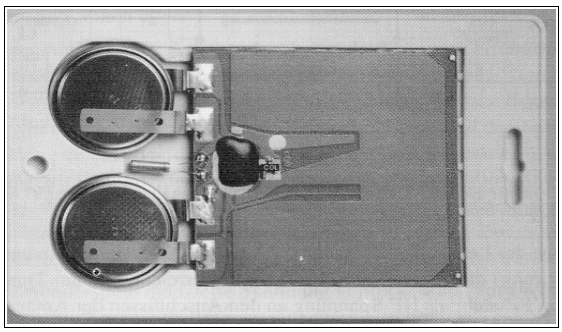
Abbildung 2.2c: Aktiver Transponder (geöffnet)

Der Einsatzzweck und die gewünschten Reichweiten führen zu einer Unterteilung in so genannte Close-Coupling-, Remote-Coupling- und Long-Range-Systeme. Bei Close-Coupling-Systemen liegt die Reichweite zwischen 0-1cm. Diese Systeme sind kapazitiv und/oder induktiv gekoppelt und erlauben hohe Übertragungsraten. Ihre Frequenz ist meist sehr niedrig, kann aber auch bis hin zu 30 MHz reichen. Bei Remote-Coupling-Systemen nimmt man eine Reichweite von ca. 1 Meter an. Lesegerät und Transponder sind hierbei induktiv gekoppelt. Typische Übertragungsfrequenzen sind 100 bis 135 kHz, 6,75 MHz, 13,56 MHz und 27,125 MHz. Es kommt hauptsächlich Lastmodulation[1] zum Einsatz. Long-Range-Systeme bieten eine Reichweite von bis zu 10 Metern und mehr. Es handelt sich grundsätzlich um aktive Transponder, welche über das Backscatter-Verfahren[2] mit dem Lesegerät kommunizieren. Als Übertragungsfrequenzen werden hauptsächlich 915 MHz, 2,45 GHz, 5,8 GHz und 24,125 GHz verwendet. Für weitere technische Einzelheiten, wie zum Beispiel die verschiedenen Modulationsverfahren, die zum Einsatz kommen, sei die Lektüre des RFID-Handbuchs [9] von Klaus Finkenzellers empfohlen.

ISO 15693

In dieser Diplomarbeit kommen als RFID-Tags sowohl Tag-It HF-I Karten der Firma Texas Instruments als auch I-Code SL1 Karten von Philips zu Einsatz. Beide Typen von Tags halten sich an die ISO 15693 Norm, sodass das verwendete Lesegerät mit beiden auf dieselbe Weise kommunizieren kann. Unter die ISO-Norm 15693 fallen RFID-Chipkarten mit einer Reichweite von bis zu einem Meter, dessen Energieversorgung auf induktive Kopplung basiert. ISO 15693 Systeme arbeiten auf einer Frequenz von 13,56 MHz mit einer Frequenztoleranz von mindestens ± 7 kHz. Zur Datenkommunikation von Lesegerät zum RFID-Tag kommt ASK-Modulation zum Einsatz. Hier stehen als Datenrate sowohl 1,65 kBit/s als auch 26,48 kBit/s zur Auswahl. Für den umgekehrten Weg von RFID-Tag zum Lesegerät wird Lastmodulation mit Hilfsträger verwendet. Der Datenstrom kann dabei ASK- oder auch FSK-moduliert sein. Welche Modulationsart zum Einsatz kommt, wird durch das Lesegerät bestimmt, indem es ein Steuerbit im Header seiner Nachricht an das Tag setzt. Mittels eines weiteren Steuerbits kann das Lesegerät auch zwischen zwei Datenraten für die Antwort des RFID-Tags auswählen. Die „long distance mode“ Datenrate beträgt 6,62 kBit/s und die „fast mode“ Datenrate 26,48 kBit/s. Die Norm schreibt für die Tags eine Seriennummer mit 64 Bit Länge vor.

1 Bei der Lastmodulation sind die Antennenspule des Lesegeräts und die Antennenspule des RFID-Tags wie ein Transformator induktiv gekoppelt. Der resonante RFID-Tag entzieht dem magnetischen Wechselfeld der Lesegerätantenne Energie. Durch das Ein- und Ausschalten eines Lastwiderstandes an der Antenne des RFID-Tags, kann die entzogene Energie verändert werden. Je nach entzogener Energie stellt sich an der Antenne des Lesegerätes eine andere Spannung ein. Durch eine Amplitudenmodulation der Spannung in der Antenne des Lesegeräts kann so die Information, die der RFID-Tag zum Lesegerät senden möchte, transportiert werden.

2 Das Backscatterverfahren basiert darauf, dass elektromagnetische Wellen von Materie reflektiert werden (zum Beispiel bekannt aus der Radar-Technik). Wie viel reflektiert wird, hängt von dem so genannten Rückstrahlquerschnitt des Objektes ab. Wenn ein Objekt in Resonanz mit der empfangenen Frequenz ist, ist der Rückstrahlquerschnitt besonders hoch. Der RFID-Tag kann durch einen parallel zu seiner Antenne liegenden, an- und abschaltbaren Lastwiderstand den Rückstrahlquerschnitt variieren, womit er amplitudenmoduliert dem Lesegerät antworten kann.

Antikollisionsverfahren der RFID-Tags

Für den Fall, dass mehrere RFID-Tags gleichzeitig im Wirkungsbereich eines Lesegeräts sind, gibt es eine breite Auswahl an Antikollisionsverfahren, die nur teilweise durch Normung geregelt sind. Die hier verwendeten Tags bieten im Rahmen der ISO 15693 an, dass das Lesegerät adressierte Befehle zusammen mit einer Bitmaske aussenden kann. Die Bitmaske bestimmt, welche Bits der Seriennummer in den RFID-Tags ausgewertet werden sollen. Zu Anfang, wenn das Lesegerät nicht weiß, welche Tags in Reichweite befindlich sind, werden durch die Bitmaske möglichst wenige Adressbits durch die Tags ausgewertet. Sollte sich nun mehr als ein Tag angesprochen fühlen, würden diese gleichzeitig auf die Anfrage des Lesegeräts antworten. Diese Kollision kann das Lesegerät bitgenau erkennen und würde im nächsten Schritt die Bitmaske und die gesetzten Bits derart verändern, dass mehr Bits der Seriennummer relevant werden. So kann sich das Lesegerät langsam an eine einzelne Seriennummer herantasten. Für die weitere Suche nach gültigen Seriennummern im Lesebereich können dann die bereits gefundenen Seriennummern ausgeschlossen werden. Ein weiteres Antikollisionsverfahren, welches ISO 15693 bereithält, ist Slotted-Aloha. Hierbei sendet das Lesegerät eine Aufforderung an alle Tags im Lesebereich, dass sie sich mit ihrer Seriennummer melden sollen. Für die Antwort stehen mehrere Zeitschlitze zur Verfügung. Die einzelnen Tags entscheiden selber, in welchem Zeitschlitz sie Antworten. Wenn mehrere Tags im selben Zeitschlitz antworten, wird keine ihrer Seriennummern erkannt. Wenn jedoch in einem der Zeitschlitze nur ein Tag antwortet, wird diese Seriennummer empfangen. Daraufhin wird das Tag mit dieser erkannten Seriennummer stumm geschaltet und es wird mit den anderen Tags, wie gerade beschrieben, weiter verfahren, bis alle Seriennummern im Lesebereich bekannt sind. Wenn sich gegenüber den zur Verfügung stehenden Zeitschlitzen relativ vielen Tags im Lesebereich befinden, kann es jedoch vorkommen, dass es sehr lange dauert, bis mal nur ein einzelnes Tag alleine in einem Zeitschlitz antwortet. Im Worst-Case-Fall antworten sogar immer in allen Zeitschlitzen mehrere Tags gleichzeitig, sodass man dann auf das vorher beschriebene Verfahren mit der bitgenauen Kollisionserkennung und der Verwendung von Bitmasken zurückgreifen muss.

Kryptographie

Auf dem Markt sind bereit RFID-Systeme mit gegenseitiger Authentifizierung und/oder verschlüsselter Datenübertragung erhältlich. Die bereits angesprochenen Tag-It HF-I und I-Code SL1 Karten bieten aber leider nichts derartiges an. Leider hat sich bisher auch noch keine Normung für die Nutzung kryptographischer Verfahren im RFID-Umfeld etabliert können.

2.3 Andere Tracking- und Ortungssysteme

Im Gegensatz zu dem hier zu entwickelnden Tracking-System, welches RFID-Tag-Landmarken zur Positionsabschätzung einsetzen soll, gibt es bereits eine Vielzahl anderer Tracking-Systeme, welche nicht auf RFID basieren. Um ein Tracking durchführen zu können, muss auf mindestens ein Ortungssystem zurückgegriffen werden. Einige Ortungssysteme sind für den großflächigen bis weltweiten Einsatz gedacht, sind dabei aber relativ teuer, ungenau oder leiden unter „Versorgungslöchern“. Andere Systeme spielen nur auf kleineren Arealen wie geschlossenen Räumen oder Lagerhallen eine Rolle, können dafür aber dort zum Beispiel mit einer hohen Präzision und Verfügbarkeit aufwarten. Sie alle haben gemeinsam, dass sie sich darin unterscheiden lassen, ob entweder von außen die Position eines beweglichen Objektes bestimmt wird oder aber das bewegliche Objekt selber seine Position bestimmen kann. In einzelnen Fällen ist auch beides gleichzeitig möglich.

Hier einige Beispiele für verwendbare Ortungssysteme:

Position des beweglichen Objektes wird durch das Objekt selber bestimmt		**Position des beweglichen Objektes wird von außen bestimmt**
GPS/DGPS		GSM Cell-ID
Loran-C		Active Bat
Funkfeuer		Active Badge
GSM Cell-ID		
Cricket		

GPS (Global Positioning System)

GPS ist ein satellitengestütztes Navigationssystem, welches vom Verteidigungsministerium der USA betrieben wird. Es ist für den weltweiten Einsatz im Freien vorgesehen. Die Nutzung in Gebäuden ist nicht möglich. Seit der Abschaltung einer künstlichen Ungenauigkeit für zivile Nutzer sind Messungen von zumeist besser als 10 Metern Genauigkeit möglich. Die für zivile Nutzer gesendeten Signale tragen keine kryptographische Signatur oder ähnliches, sodass das Senden von gefälschten

Satellitensignalen (Spoofing) möglich wäre[1] [2]. Die US-Streitkräfte können sich vor so einen Angriff schützen, indem zusätzlich ein weiteres Signal, der so genannte P-Code, welcher in verschlüsselter Form von den Satelliten ausgesendet wird, ausgewertet wird. Zusätzlich leistet der P-Code, dass im Kriegsfall künstlich verschlechterte Satellitensignale von berechtigten Nutzern wieder für eine korrekte Positionsbestimmung verwendet werden können.

Abbildung 2.3a: Hausarrest durch GPS-Tracking
(siehe Fußzeile)

DGPS (Differential Global Positioning System)

DGPS ist ein Sammelbegriff für verschiedene Verfahren, in denen mehrere GPS-Empfänger zum Erreichen einer höheren Genauigkeit gleichzeitig verwendet werden. Hierbei ist die Position einer oder mehrerer Basisstationen genauestens bekannt. Die Basisstationen sind in der Lage, fehlerhafte GPS-Positionsdaten zu analysieren, die zum Beispiel aus Störungen durch die Ionosphäre hervorgerufen werden. Die durch die Analyse berechneten Korrekturdaten werden an mobile GPS-Empfänger gesendet, welche damit dann für sich wiederum ihre tatsächliche Position um einiges genauer bestimmen können als ohne diese Korrekturdaten. Je nach DGPS-System sind die erzielbaren Genauigkeiten noch einmal sehr unterschiedlich:

1 Sollte zum Beispiel eine Person mit Hausarrest eine Manschette mit GPS-Empfänger tragen [10], mit der er in regelmäßigen Abständen ins Freie treten soll, um seine Position bestimmen zu lassen, wäre eine Manipulation der Positionsbestimmung denkbar. Die Manipulation bestünde darin die Manschette vor den eigentlichen GPS-Signalen abzuschirmen, um stattdessen selbst generierte Funksignale an den Manschettenempfänger zu senden, die daraufhin zu einer falschen Positionsberechnung führen.

2 Dieses würde es einem Gefangenen mit GPS-Hausarrest-Manschette ermöglichen, sich an anderen Orten aufzuhalten, als es ihm erlaubt ist. Allgemein gesprochen ist also anzumerken, dass, wenn erwogen wird für eine Anwendungen GPS einzusetzen, auch daran gedacht werden sollte, dass ein Angreifer unbemerkt dafür sorgen könnten, dass falsche Positionsbestimmungen durchgeführt werden.

Eurofix	Bis 200 Seemeilen Entfernung vom Sender weniger als 3 Meter Abweichung in 95% der Messungen (verwendet werden alten Loran-C-Sender zur DGPS-Korrekturdaten-aussendung) (Real-Time)
SAPOS EPS	0,5 bis 3 Meter Genauigkeit (Sender: UKW-Radiosender der ARD, Langwellensender der Telekom und Sender der Landesvermessung z.B. in Hamburg: 160,31 und 161,07 MHz) (Real-Time)
SAPOS HEPS	1-2 cm Genauigkeit (Sender: im 2-Meter-Band und über Modem per Telefonnetz) (Real-Time)
SAPOS GPPS	1cm Genauigkeit. Korrekturdaten werden auf einem Datenträger geliefert, sind per Mailbox abrufbar und sind zum Teil per ftp im Internet erhältlich. (Post-Processing innerhalb von etwa 15 Minuten).
SAPOS GHPS	Unter 1cm Genauigkeit (Post-Processing)
EGNOS	1-3 Meter Genauigkeit - Sender: geostationärere Satelliten mit 34 Referenzstationen am Boden

Beim „SAPOS EPS“ des Satellitenvermessungsdienstes der deutschen Landesvermessung [11] werden die Korrekturdaten unter anderem durch das patentierte „RASANT“-Verfahren[1] ausgesendet. Hierbei werden die Differential-GPS-Korrekturdaten im RDS-Signal von Radiosendern der ARD transportiert.

Genauigkeitsvergleich zwischen GPS und RASANT-DGPS:

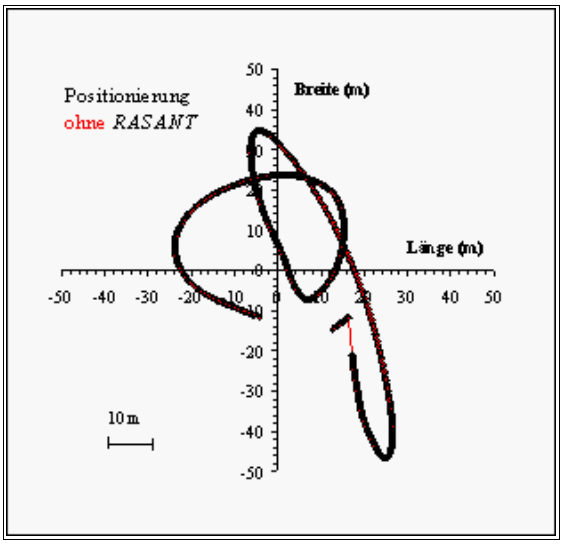

Abbildung 2.3b: Autonomes GPS mit SA

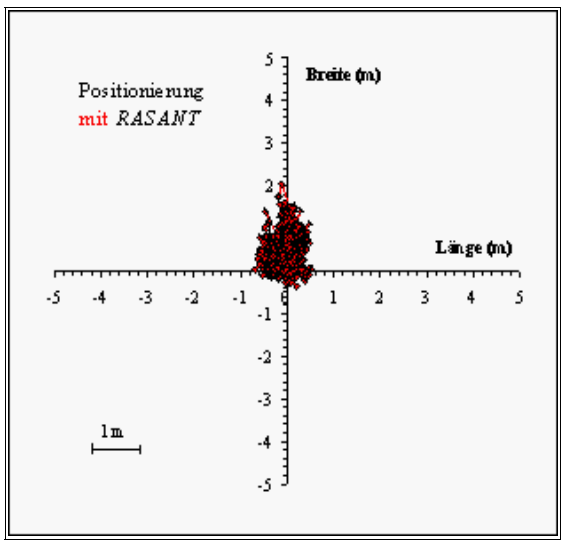

Abbildung 2.3c: DGPS mit RASANT

1 Patent für RASANT beim europäisches Patentamt EP0847537 [12], US-Patent 6.018.313 [13] [14] (siehe Anhang)

Loran-C

Loran-C ist ein altes Funk-Navigationssystem aus der Schifffahrt, welches durch die neu hinzugekommene Mitbenutzung der Senderketten als DGPS-Sender im Eurofix-System wohl doch noch länger im Betrieb bleiben wird, als vor einigen Jahren noch angenommen wurde. Loran-C gehört zu den Hyperbelnavigationsverfahren. Eine Loran-C-Senderkette besteht aus einem Hauptsender und zwei bis vier Nebensendern. Die Position lässt sich mit Loran-C auf etwa 30 Meter genau bestimmen. Eine genaue Beschreibung des Verfahrens ist unter [15] zu finden.

Funkfeuer

Es wird zwischen gerichtetem und ungerichtetem Funkfeuer unterschieden. Ungerichtetes Funkfeuer wird sowohl in der Schifffahrt (nur noch selten), wie auch in der Flugnavigation eingesetzt. Gerichtete Funkfeuer finden vor allem in der Flugnavigation Anwendung. Als Beispiel für ein gerichtetes Funkfeuer-Verfahren sei hier folgend das UKW-Drehfunkfeuer VOR in kurzen Worten beschrieben.

Abbildung 2.3d: VOR Anlage

VOR ist die Abkürzung für „VHF Omni Range“ und ist im Frequenzbereich von 108 bis 118 MHz zu finden. Man kann sich ein gerichtetes Funkfeuer ähnlich wie einen Leuchtturm vorstellen. „Zum besseren Verständnis folgendes Denkmodell: Ein Leuchtturm strahlt jeweils zur vollen Minute ein kurzes Lichtsignal gleichmäßig in alle Richtungen ab. Gleichzeitig startet ein horizontal umlaufender, eng gebündelter Lichtstrahl, der für einen vollen Umlauf 360 Sekunden benötigt, d.h. ein Grad pro Sekunde. Ein Schiffsführer, der 135 Sekunden nach dem Aufblitzen des Lichtes an der Leuchtturmspitze den Peilstrahl sieht weiß, dass er sich auf der Kurslinie 135° befindet.“ (zitiert aus [16]). Wer sich nicht mit diesem Denkmodell begnügen möchte, sondern etwas genauer wissen will, wie VOR funktioniert, sollte einen Blick auf die englischsprachige Version von Wikipedia zum Begriff „VOR“ [17] riskieren.

Abbildung 2.3e: UKW-Drehfunkfeuer Elbe auf 115.10 MHz

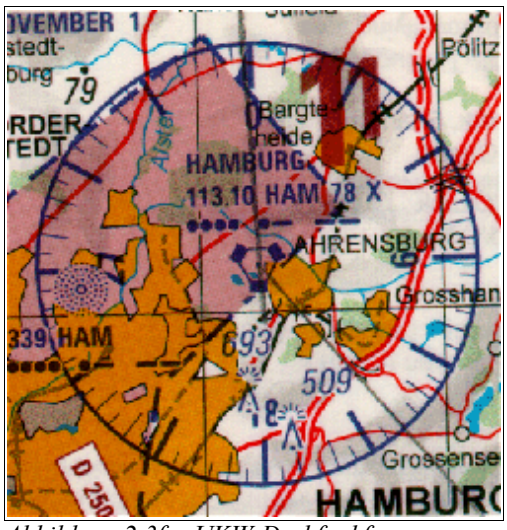

Abbildung 2.3f: UKW-Drehfunkfeuer Hamburg auf 113.10 MHz

GSM

Eine Möglichkeit mit Hilfe von GSM eine Positionsbestimmung durchzuführen, besteht in der Auswertung der Cell-ID der Zelle, in der sich das Gerät momentan befindet. Abhängig von der Größe der Funkzelle von wenigen hundert Metern bis mehreren Kilometern, ist eine mehr oder weniger genaue Positionsbestimmung möglich.

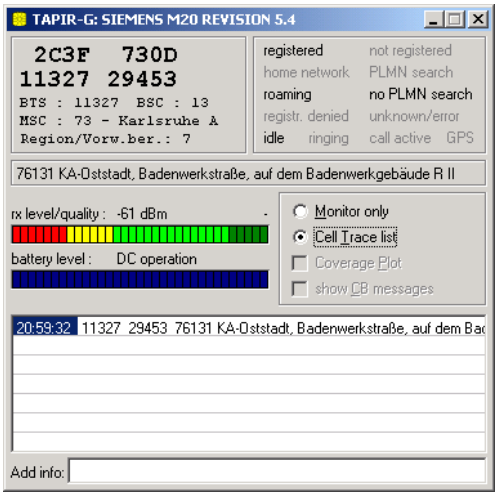

Abbildung 2.3g: Monitorsoftware (oben Links steht die Cell ID)

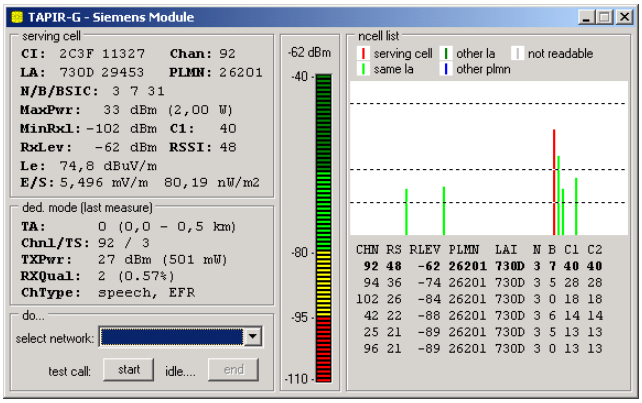

Abbildung 2.3h: Monitorsoftware (oben Links steht die Cell ID)

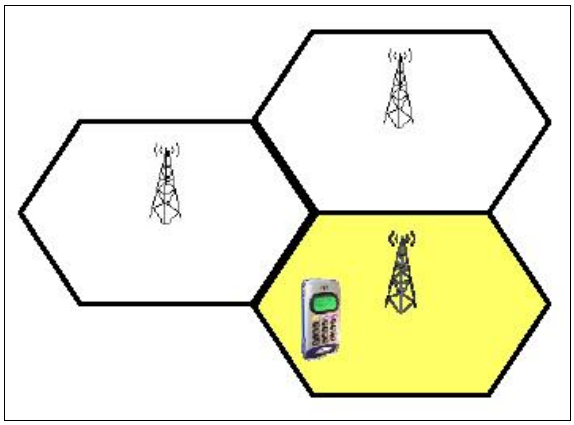

Abbildung 2.3i: Funkzellen

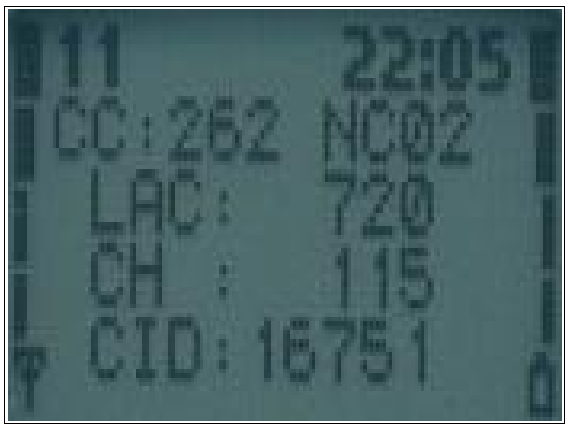

Abbildung 2.3j: Netmonitor eines Nokia Mobiltel. „CID“ steht für „Cell ID“

Cricket

Cricket ist ein Verfahren, welches zur Positionsbestimmung innerhalb von Gebäuden benutzt werden kann. Hierbei gibt es Landmarken, welche zeitgleich ein Funk- und ein Ultraschallsignal aussenden. Der Empfänger kann aus der zeitlichen Differenz der Empfangszeitpunkte beider Signale den Abstand zu einer Landmarke bestimmen. Wenn der Empfänger auf diese Weise seine Entfernung zu mehreren Landmarken bestimmt hat, kann er aus diesen ermittelten Daten über Triangulation seine Position im Raum errechnen. Auf der Cricket-Projekthomepage des Massachusetts Institute of Technology [18] findet sich unter anderem die Information, dass mit diesem System eine Genauigkeit von 1-3 cm erreicht werden kann. Wer Interesse an einem Nachbau hat, findet auf der Projekthomepage auch alle Schaltpläne und die passende Software dazu.

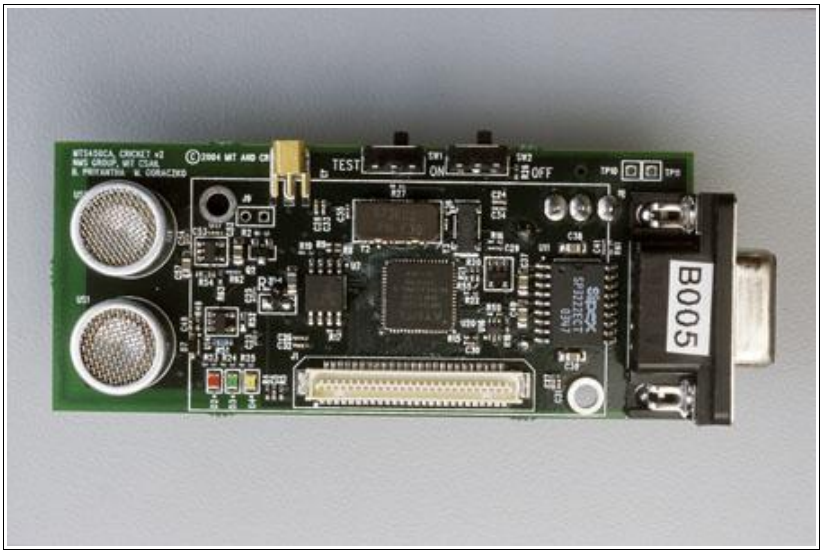

Abbildung 2.3k: Cricket-Hardware vom MIT (vereint sowohl Empfänger zur Positionbestimmung als auch Beacon/Bake)

ActiveBat

Auch ActiveBat ist für den lokalen Einsatz in Gebäuden konzipiert. Hier trägt das zu verfolgende Objekt ein Gerät namens „Bat“ (Fledermaus) bei sich. In regelmäßigen Abständen sendet ein Zentralrechner über Funk an die „Bat“ den Befehl, dass diese ein Ultraschallsignal aussenden soll. Durch Laufzeitmessungen zu an der Decke befindlichen Sensoren wird durch den Zentralrechner die aktuelle Position der „Bat“ berechnet. Nach einer Untersuchung im Rahmen des Programms MobilMedia des Bundesministeriums für Wirtschaft und Arbeit [19] kann eine Ortungsgenauigkeit von etwa 10cm erreicht werden, wenn an der Decke ein Sensoren-Netzwerkraster von 1,20 Metern eingesetzt wird.

Active Badge Location System

Bei Active Badge trägt das zu trackende Objekt eine Badge (Marke), welche in regelmäßigen Abständen über Infrarotsignale seine ID aussendet. Empfänger in den jeweiligen Räumen des Gebäudes leiten diese Information dann zum Beispiel an einen Zentralrechner weiter, welcher auf diese Wiese dann über den Aufenthaltsort der einzelnen Badges informiert ist. Die Ortungsgenauigkeit ist hierbei natürlich systembedingt relativ gering.

Infrarot-Baken-System im öffentlichen Personennahverkehr

Siemens VDO Automotiv geht im Vergleich zu Active Badge im Rahmen ihres integrierten Funkanforderungssystems IFAS [20] den genau umgekehrten Weg. Bei dem Siemens-System werden in einigem Abstand vor Ampelanlagen Infrarot-Baken aufgestellt, welche auf diese Weise als künstliche Landmarken für die Busse dienen. Wenn nun ein Bus an so einer Landmarke vorbeifährt, kann er dadurch seine aktuelle Position erkennen. Basierend auf dieser Information kann der Bus nun der vor ihm liegende Ampelanlage per Funk ein Datentelegramm übermitteln, welches eine Grünphase für ihn einleitet[1].

1 Zum Einsatz kommt dieses Infrarot-Baken-System zum Beispiel im Norderstedter Bereich des HVV. Hier verwenden die Busse der HHA dieses System der Positionsbestimmung [21]. Da der HHA mittlerweile zum besseren Management ihres Fuhrparks die Bordrechner der einzelnen Fahrzeuge durch (D)GPS-Empfänger erweitert hat [22], ist zu vermuten, dass für den Fall einer Ausweitung der LSA-Beeinflussung (LSA = Lichtsignalanlage) auf das gesamte HVV-Gebiet eher keine weiteren Infrarot-Baken aufgestellt werden.

2.4 RFCOMM (Bluetooth)

Das RFCOMM Protokoll ist Bestandteil des in der Bluetooth-Spezifikation beschriebenen Protokollstacks (siehe Abb. 2.4a). Durch RFCOMM ist man in der Lage, die serielle Verbindung einer RS232-Schnittstelle zu emulieren. Es können alle neun Leitungen inklusive der Steuersignale übertragen werden[1]. Das Protokoll sieht ein Maximum von 60 gleichzeitigen RS232-Verbindungen zur tiefer liegenden L2CAP-Protokollschicht vor. Die tatsächliche maximale Anzahl hängt jedoch von der konkreten Implementationen innerhalb der verwendeten Bluetoothgeräte ab. Weiterführende Informationen zu Bluetooth und seinen Protokollen findet man zum Beispiel beim „palowireless – Wireless Resource Center“ [24].

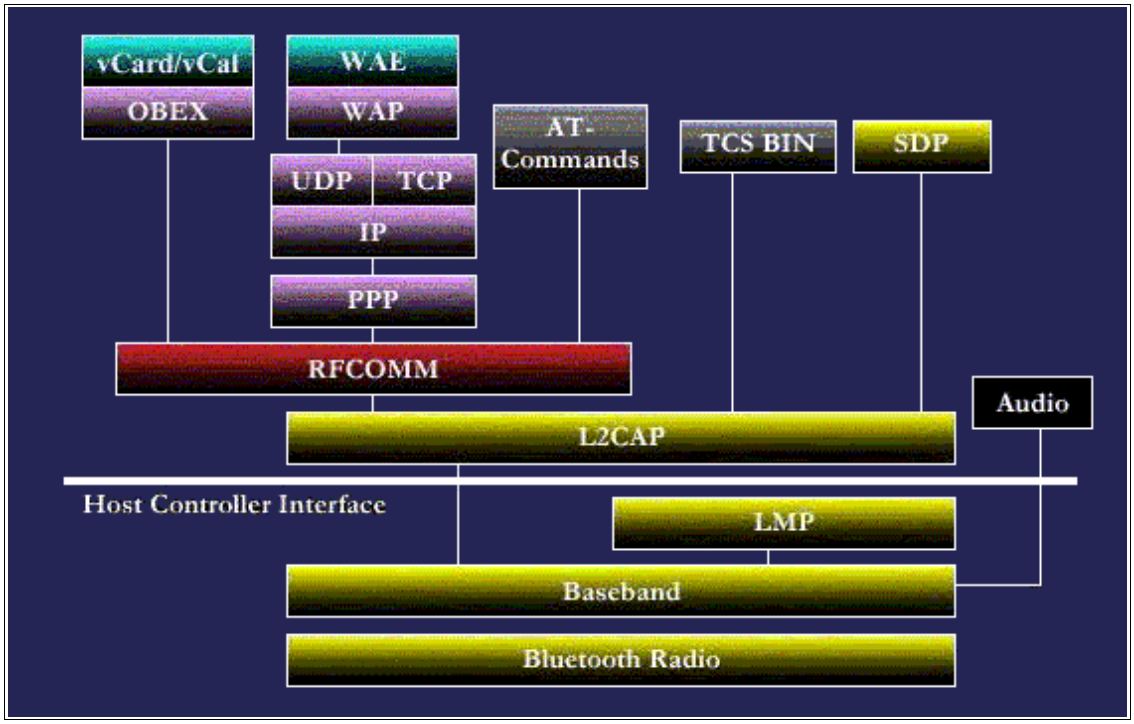

Abbildung 2.4a: Bluetooth Specification Protocol Stack

1 Im Serial Port Profile ist die Implementation der Steuersignale (bis auf die Flusskontrolle) optional. Siehe auch: [23]

Kapitel 3

Analyse

Wie bereits im Kapitel 1.2 angesprochen wurde, gibt es zwei Typen von Aufgaben innerhalb des hier zu entwickelnden Tracking-Systems: fortwährende und nur bei Bedarf anstehende Aufgaben. Zum einen soll durch die „RFID-Watch-Box“ eine fortwährende Überwachung der Umgebung nach auftauchenden und wieder verschwindenden RFID-Tags stattfinden, wobei diese Ereignisse protokolliert werden sollen. Zudem soll die „RFID-Watch-Box“ im Rahmen der fortwährenden Überwachung im Falle einer Verletzung von zuvor aufgestellten Regeln ein akustischer Alarm ausgeben. Zum anderen soll der so genannte „RFID-Configurator“ im Unterschied zu den fortwährenden Aufgaben dem Benutzer nur bei Bedarf eine Menüführung zur Verfügung stellen, die es ihm erlaubt, auf die gesammelten Daten zuzugreifen, Regeln und RFID-Tags zu verwalten, sowie nähere Informationen zu eingetretenen Regelverletzungen abzufragen. Die Aufteilung in fortwährende sowie nur vereinzelt und kurzzeitig auftretende Aufgaben bietet an, auch dieses Kapitel zur Analyse in gleicher Weise aufzuteilen:

- **Kapitel 3.1** Analyse des „RFID-Configurators“ (bei Bedarf anstehenden Aufgaben)
- **Kapitel 3.2** Analyse der „RFID-Watch-Box“ (fortwährenden Aufgaben)

In diesen beiden Abschnitten soll im Rahmen der Anforderungsanalyse auf folgende Punkte eingegangen werden:

1. Festlegung der Anforderungen
2. Spezifikation der Anwendungsfälle
3. Prototypen einer möglichen Benutzeroberfläche

Dazu werden die Anforderungen in Textform erarbeitet. Daraus werden dann im nächsten Schritt die Anwendungsfälle abgeleitet, welche zum besseren Verständnis durch Use-Case-Diagramme übersichtlich dargestellt werden. Alsdann werden schließlich noch Screenshots von Prototypen einer möglichen Benutzeroberfläche gezeigt.

In Kapitel 3.3 erfolgt zusätzlich noch ein kurzer Exkurs zum Thema Sicherheitsanalyse im Sinne der Informationsvertraulichkeit. Dabei soll direkter Bezug auf das hier zu entwickelnden Tracking-System auf Basis von RFID genommen werden. Neben der in der Softwaretechnik üblichen Betrachtungsweise dieses Themas sollen in Kapitel 3.3 auch Aspekte der Elektro- beziehungsweise Nachrichtentechnik angesprochen werden, die in diesem Zusammenhang interessant sein könnten.

3.1 Analyse des „RFID-Configurators“

3.1.1 Festlegung der Anforderungen

Verwaltung von RFID-Tags

Dem Benutzer des zu entwickelnden Trackingsystems kann es selbstverständlich nicht zugemutet werden, dass er sich immer und immer wieder mit den für Ihn kryptisch anmutenden Identifikationnummern der RFID-Tags konfrontiert sieht. So muss das Trackingsystem dem Benutzer die Möglichkeit bieten, den Identifikationsnummern jeweils einen selbst gewählten Namen und einen weiterführenden und möglichst gut beschreibenden Text zuzuordnen.

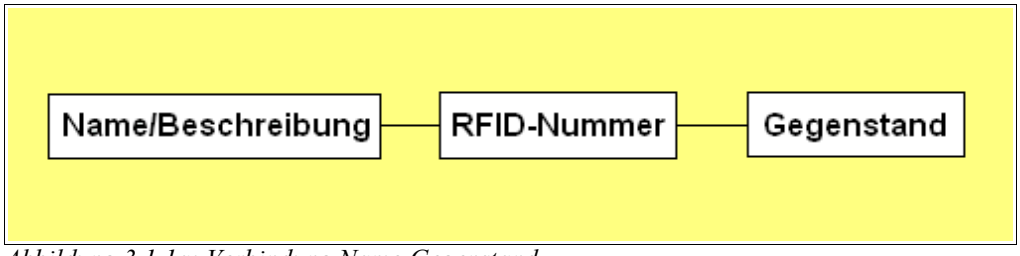

Abbildung 3.1.1a: Verbindung Name-Gegenstand

Da die RFID-Tags mit den ihnen zugeordneten Gegenständen dauerhaft verbunden sein sollen, können wir die Beziehung von Gegenstand zu RFID-Nummer als eindeutig und invariant ansehen. Darüber hinaus also auch die Beziehung von einem selbst gewählten

Namen beziehungsweise einer selbst gewählten Beschreibung zu dem jeweiligen über RFID-Tag markierten Gegenstand (siehe Abb. 3.1.1a).

Neben dem Namen und der Beschreibung soll dem jeweiligen Gegenstand auch noch ein Typ zugewiesen werden können. Ein Gegenstand kann zu einer von drei Typausprägungen gehören: Objekt, Person oder Landmarke. Eine Landmarke hat die Eigenschaft, einen bestimmten Ort zu definieren. Ein Beispiel sei hier „Fahrstuhleingang im 11.Stock im E+I-Gebäude“. Im Gegensatz zum Objekt wissen wir immer genau wo sich eine Landmarke befindet. Hiermit haben wir auch gleich einen der wichtigsten Eigenschaften eines Objektes: Ein Objekt kann sich im Gegensatz zu einer Landmarke auch mal woanders aufhalten – es ist also ortsvariabel. Das heißt, dass wir auch mal auf die Idee kommen könnten, nach einem Objekt suchen zu müssen, da wir nicht mehr wissen, wo wir es zuletzt gesehen haben. Als Beispiel für ein Objekt sei hier „mein PDA“ genannt[1]. Schließlich gibt es noch Personen, welche vom Grundsatz her viel Ähnlichkeit mit Objekten haben, aber nicht nur vom ethischen Standpunkt aus gesehen von diesen unterschieden werden sollten. Zur technischen Notwendigkeit der Unterscheidung zwischen Personen und Gegenständen wird im Kapitel 5.2.5 tiefer eingegangen. Hier sei erstmal darauf vertraut, dass der Benutzer des Systems von sich aus eine Person von einem beliebig anderen Objekt unterscheiden kann.

Die RFID-Tag-Verwaltung hat auch die Aufgabe, dem Benutzer die von der Systemkomponente „RFID-Watch-Box“ gesammelten Informationen über die jeweiligen mit RFID-Tag markierten Gegenstände bei Bedarf anzuzeigen. Die gesammelten Informationen beinhalten den Zeitpunkt der letzten Sichtung, die letzte Landmarke vor der letzten Sichtung, wie auch die erste Landmarke nach der letzten Sichtung. Die Information über diese beiden Landmarken soll auch den jeweiligen Zeitpunkt dessen Sichtung beinhalten. Es soll zwei Arten einer RFID-Tag-Übersicht geben. Zum einen ist eine Übersicht aller aktuell in Reichweite befindlichen RFID-Tags vorzusehen, zum anderen soll auch eine Gesamtübersicht über alle bekannten und zu trackenden RFID-Tags vorhanden sein.

Verwaltung von Regeln

Nun kommen wir zu den aufzustellenden Regeln, welche definieren sollen, wann das Tracking-System Alarm schlagen soll und wann nicht. Nehmen wir hier zum Beispiel eine Regel, die wir in natürlicher Sprache wohl mit den Worten „meinen PDA nicht im Büro liegen lassen“ bezeichnen würden. Eine dazu passende Regel für das Tracking-System müsste heißen (ebenfalls in natürlicher Sprache formuliert): „Wenn die RFID-Landmarke für den Ausgang des Bürogebäudes gesehen wird und gleichzeitig nicht die RFID-

1 Anmerkung: Etwas problematisch stellen sich in diesem Zusammenhang Gegenstände wie Autos dar, welche eine Art ortsvariable Landmarke, aber auch Objekte sind. Dazu zur Verdeutlichung zwei Sätze: „meinen PDA habe ich zuletzt im Auto gesehen“ und „Wo habe ich zuletzt mein Auto gesehen?“

Objektkennung für meinen PDA in Reichweite ist, muss ein Alarm erfolgen“.

Die Kurzschreibweise für diese Regel könnte wie folgt aussehen:

+ Landmarke.Bürogebäudeausgang

─ Objekt.meinPDA

► Alarm: „Achtung! Sie haben Ihren PDA nicht bei sich!“

Auch hier sollte es dem Benutzer möglich sein, beim Anlegen der Regel dieser noch einen Namen und einen weiteren beschreibenden Text zuzuordnen, wie es auch schon beim Verwalten der RFID-Tags vorgesehen sein sollte. Dieses ist auch hier wünschenswert, da beim späteren Verwalten der Regeln nicht vorausgesetzt werden kann, dass der Benutzer auf Anhieb noch weiß, wofür die Regel „+Landmarke.Büroausgang ─Objekt.meinPDA“ stehen soll. Auch kann nicht vorausgesetzt werden, dass die mit angegebene Alarmmeldung hierfür ausreichend Information bietet.

Dem aufmerksamen Leser mag jetzt aufgefallen sein, dass im Rahmen der soeben aufgestellten Regel ein Alarm sowohl beim Verlassen, wie auch beim Betreten des Gebäudes erfolgt, sofern wir unseren PDA nicht bei uns tragen. Ursprünglich wollten wir jedoch, dass nur ein Alarm ausgelöst wird, wenn wir das Büro ohne PDA verlassen. Um dieses Problem zu lösen, bräuchten wir aber ein zustandsabhängiges Regelsystem, welches auch Regeln wie folgende akzeptieren würde: „Wenn wir zuerst die Landmarke Bürogebäudeausgang-Empfangshalle und danach die Landmarke Bürogebäudeausgang-VorDemGebäude sehen und gleichzeitig mit dem Sehen der Landmarke Bürogebäudeausgang-VorDemGebäude nicht die RFID-Objektkennung für meinen PDA in Reichweite ist, muss ein Alarm erfolgen“. Im Rahmen dieser Diplomarbeit soll jedoch das vereinfachte Regelsystem genügen.

Die, wie soeben beschrieben, aufgestellten Regeln sollen durch den Benutzer zeitweise außer Kraft gesetzt werden können. Eine Regelübersicht soll über den Aktivitäts- bzw. Inaktivitätszustand der einzelnen Regeln Auskunft geben. Sollte eine Regel aktuell deaktiviert sein, soll auch die noch verbleibende Zeit dieser Suspendierung angezeigt werden. Zudem soll es dem Benutzer möglich sein, eine suspendierte Regel, unabhängig von der eigentlich noch verbleibenden Zeit der Deaktivierung, sofort wieder zu aktivieren.

Meldung von und Umgang mit aktuellen Regelverstößen

Regelverstöße meldet die „RFID-Watch-Box“ durch einen akustischen Warnhinweis. Dem Benutzer soll es dann möglich sein, mit Hilfe des „RFID-Configurators“ weiterführende Informationen über diesen Regelverstoß einzuholen. Diese Informationen sollen bestehen aus:

1. dem Namen der Regel, gegen welche verstoßen wurde
2. der weiterführenden Beschreibung der Regel
3. dem Zeitpunkt, zu welchem gegen diese Regel zuletzt verstoßen wurde

Zur Bestätigung, dass der Benutzer den Regelverstoß im Nachhinein akzeptiert beziehungsweise sein Eintreten zur Kenntnis genommen hat, muss der Benutzer die betreffende Regel für eine oder mehr Minuten suspendieren.

3.1.2 Ermittlung der Anwendungsfälle

Nun folgend werden die Anwendungsfälle, welche aus den zuvor aufgestellten Anforderungen hervorgehen, in Form von Use-Case-Diagrammen dargestellt.

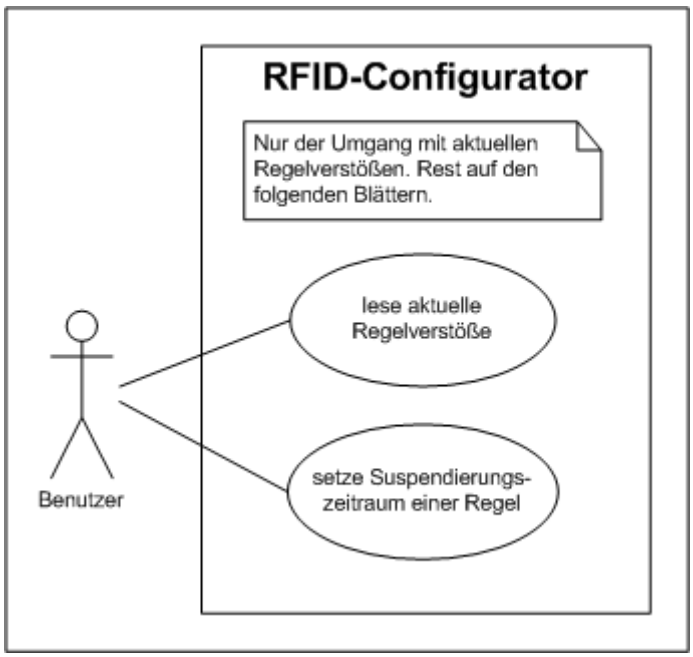

Abbildung 3.1.2a: Use-Case-Diagramm “Meldung von und Umgang mit aktuellen Regelverstößen“

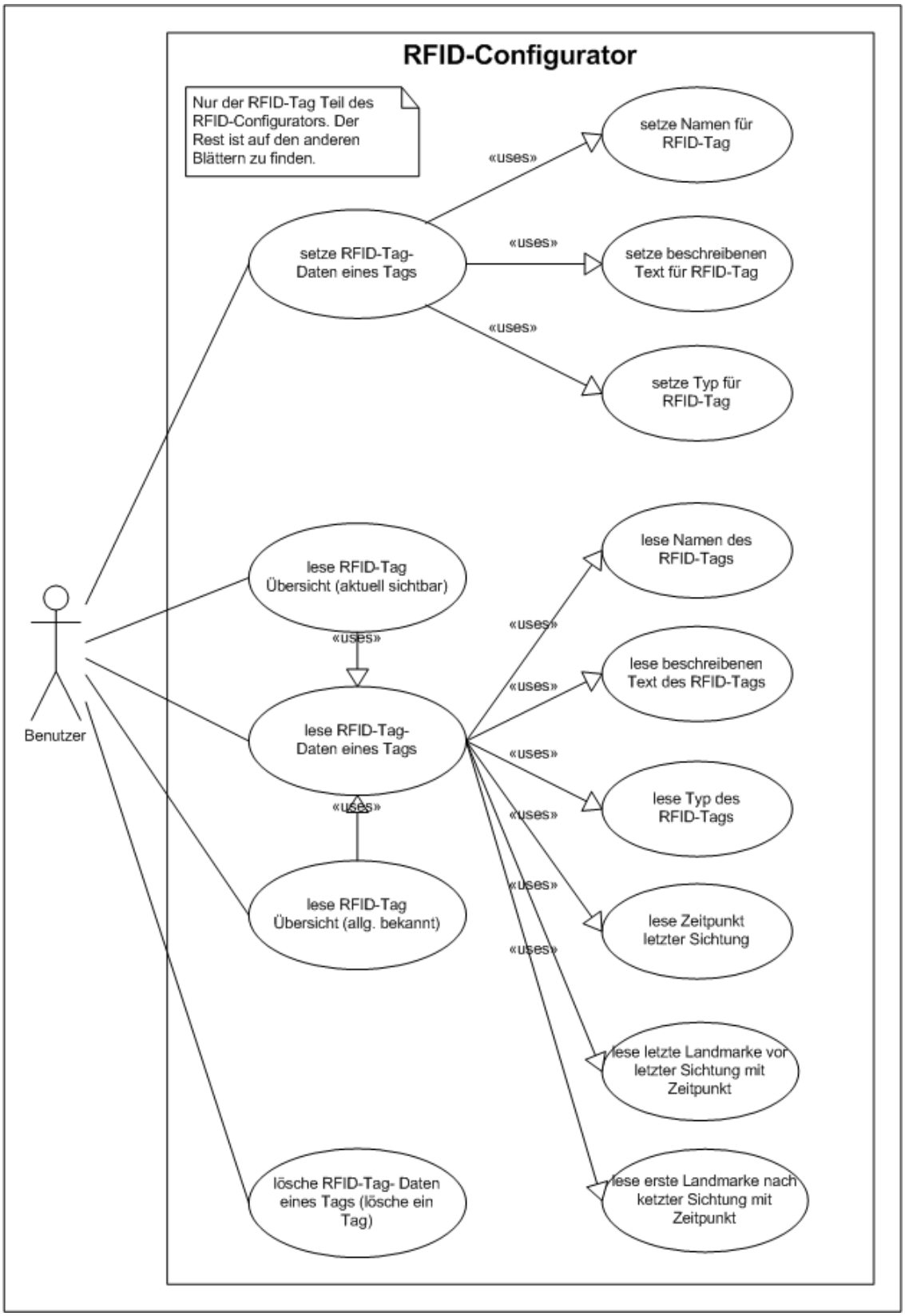

Abbildung 3.1.2b: Use-Case-Diagramm "Verwaltung von RFID-Tags"

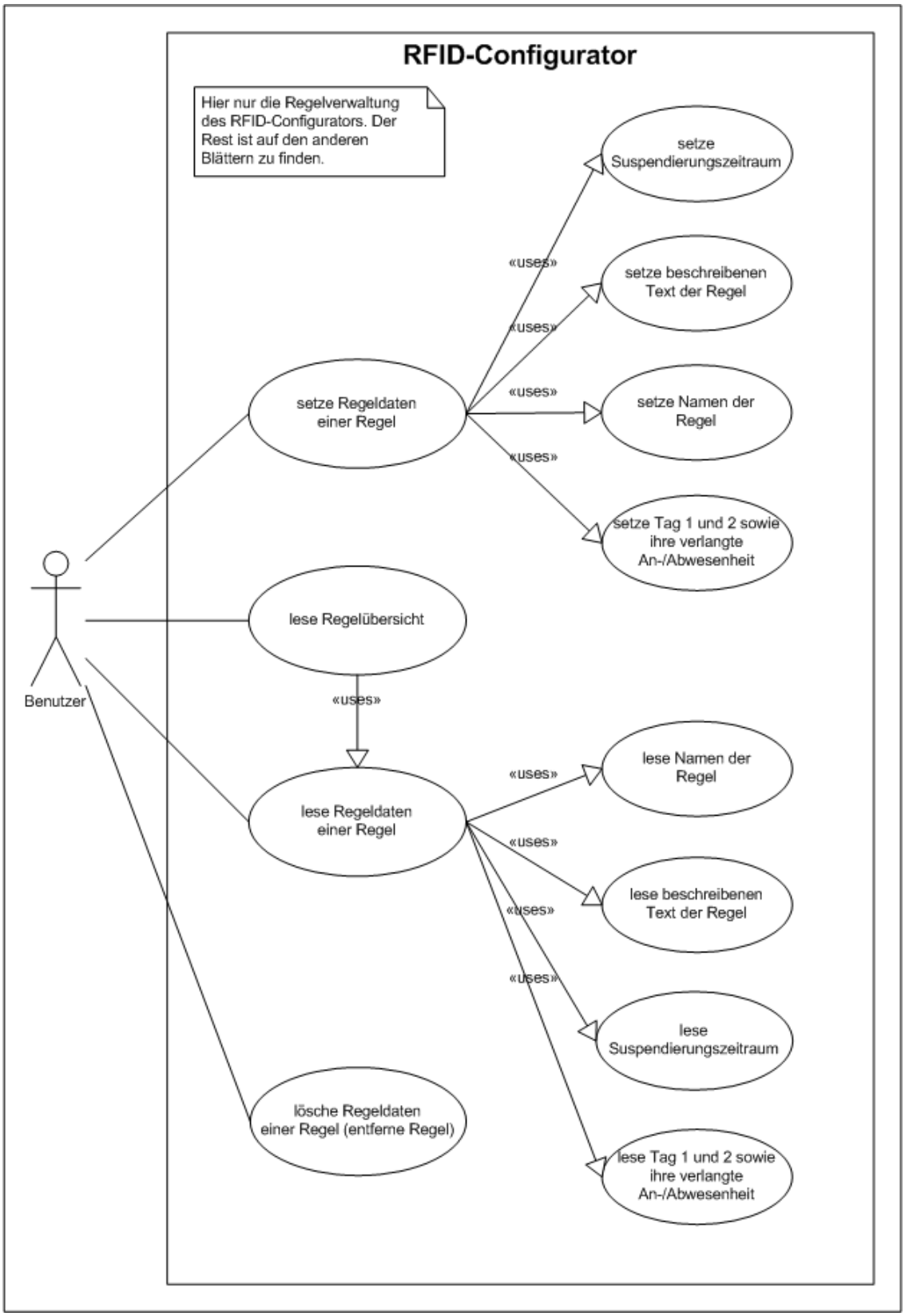

Abbildung 3.1.2.c: Use-Case-Diagramm "Verwaltung von Regeln"

3.1.3 Mögliche grafische Benutzerschnittstelle für den „RFID-Configurator“ und weitere Spezifikation der Anwendungsfälle

Für den nun folgenden Vorschlag einer möglichen Darstellungsart wird die Oberfläche der Benutzerschnittstelle in einer optimierten Form für PocketPC-PDAs gezeigt.

Das Main-Panel

Nach dem Programmstart soll dem Benutzer als erstes eine Übersicht über aktuelle Regelverstöße angezeigt werden. Sollten im Augenblick des Programmstarts keine bis dahin aufgetretenen Regelverstöße vorliegen, wird dieses ebenfalls dargestellt (siehe Abb. 3.1.3a).

Abbildung 3.1.3a: Main-Panel

Sollte es jedoch zu einem oder mehreren Verstößen gekommen sein, welche bis dahin noch nicht als im Nachhinein akzeptiert wurden, wird jeweils einer von diesen Verstößen angezeigt. Dieser Verstoß muss nun nachträglich für eine oder mehr Minuten ab dem aktuellen Zeitpunkt akzeptiert werden. Nach dem Akzeptieren wird der nächste Regelverstoß angezeigt, der ebenfalls im Nachhinein hingenommen werden muss. Dieses wiederholt sich so lange, bis alle Verstöße abgearbeitet sind. Sollte gegen eine Regel mehrfach verstoßen worden sein, wird nur der jeweils aktuellste Regelverstoß gegen diese eine Regel angezeigt und auch nur dieser Verstoß gegen diese eine Regel muss im Nachhinein akzeptiert werden.

Das Search-Panel

Wenn der Benutzer auf den Reiter „Search“ klickt, sollen zwei weitere Registerkarten mit den jeweils dazugehörigen Reitern „Tags In Range“ und „All Known Tags“ zum Vorschein kommen (siehe Abb. 3.1.3b).

Abbildung 3.1.3b: Search-Panel

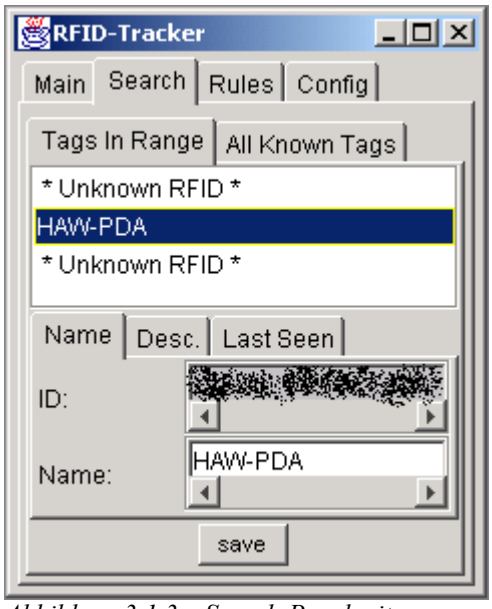

Abbildung 3.1.3c: Search-Panel mit „Konfigurationsmenü“

„Tags In Range“ soll dem Benutzer Auskunft über die aktuell im Lesebereich befindlichen RFID-Tags geben. Hier werden sowohl bekannte, wie auch unbekannte RFID-Tags angezeigt, sofern sie sich in der Reichweite des RFID-Lesegeräts befinden. Durch das Drücken auf einen dieser RFID-Tag-Namen, soll sich ein zusätzliches „Konfigurationsmenü“ öffnen. Dieses „Konfigurationsmenü“ soll die Möglichkeit bieten, bei bereits bekannten Tags die vom Benutzer änderbaren Daten zu editieren und bei unbekannten RFID-Tags dieses Daten nun zu spezifizieren, um sie Anschließend abzuspeichern (siehe Abb. 3.1.3c). Das Abspeichern eines zuvor unbekannten RFID-Tags soll zur Folge haben, dass dieses Tag von nun an zu den bekannten und zu trackenden Tags gehört.

Unter „All Known Tags“ sollen sich alle bekannten RFID-Tags finden lassen. Unter allen bekannten RFID-Tags sind diejenigen Tags zu verstehen, welche durch die „RFID-Watch-Box“ getrackt werden sollen. Die Anzeige der einzelnen RFID-Tags unter „All Known Tags“ ist unabhängig davon, ob sie sich im Moment in der Reichweite des RFID-Lesegeräts befinden oder auch nicht (siehe Abb. 3.1.3d). Auch unter „All Known Tags“ soll es möglich sein, durch Anklicken des Tag-Namens dessen „Konfigurationsmenü“ zu sehen (siehe Abb. 3.1.3e).

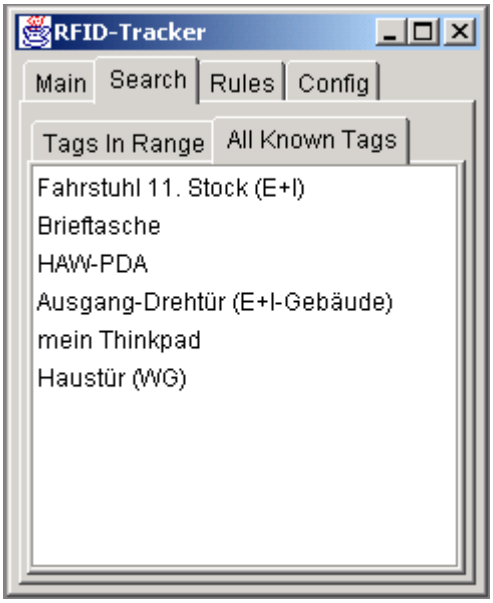

Abbildung 3.1.3d: Search-Panel "All-Known-Tags"

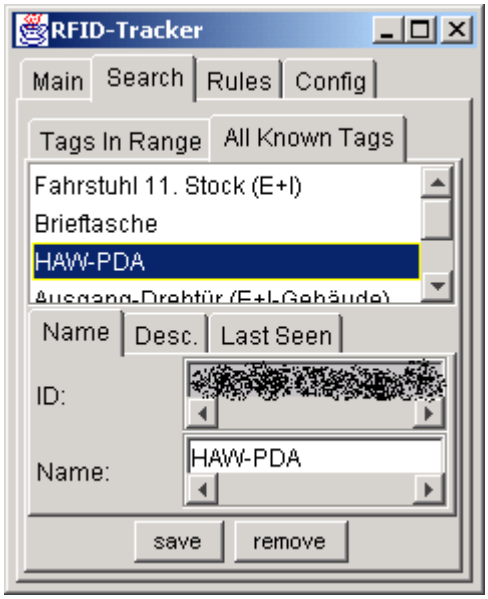

Abbildung 3.1.3e: Search-Panel mit „Konfigurationsmenü“

Wie schon erwähnt wurde, soll die RFID-Watch-Box die bekannten Tags tracken, also dessen jeweils letzte Sichtung speichern. Auch diese Information über die letzte Sichtung soll sich über das „Konfigurationsmenü“ des jeweiligen bekannten RFID-Tags in Erfahrung bringen lassen können. Dazu soll es unter „Last Seen“ die zusätzlichen Registerkarten „Time“, „Before“ und „After“ anbieten (siehe Abb. 3.1.3f).

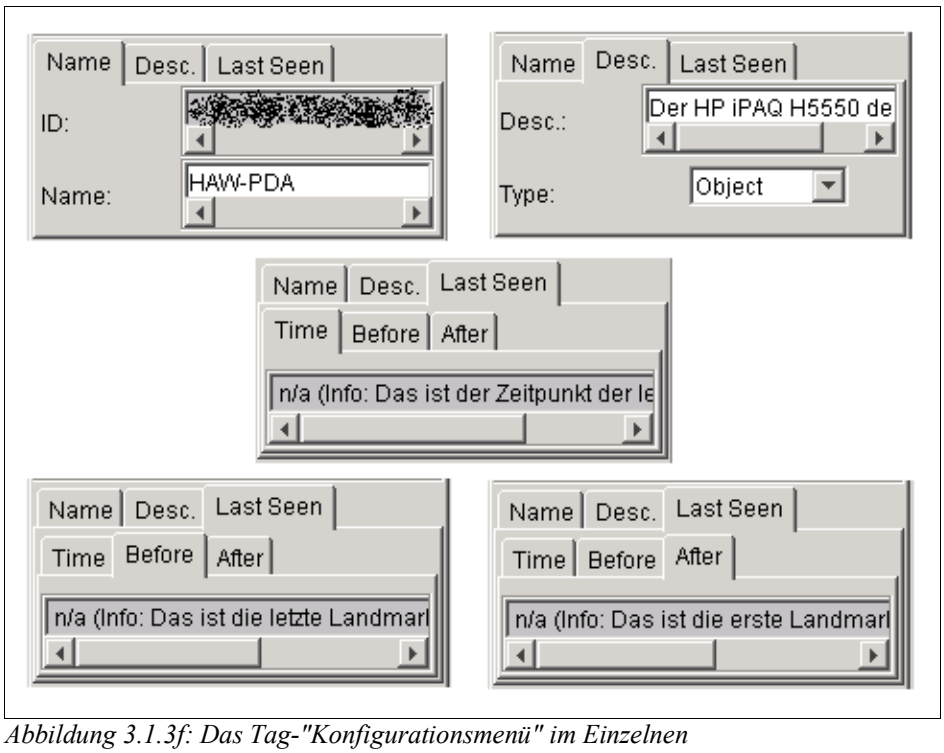

Abbildung 3.1.3f: Das Tag-"Konfigurationsmenü" im Einzelnen

Unter „Time“ soll dann der Zeitpunkt der letzten Sichtung dieses RFID-Tags verzeichnet sein, unter „Before“ die letzte Landmarke vor dem Verschwinden (inklusive Sichtungszeitpunkt der Landmarke) und unter „After“ die erste Landmarke nach dem Verschwinden (ebenfalls inklusive Sichtungszeitpunkt dieser Landmarke).

Unter „All Known Tags“ soll es auch möglich sein, bekannte RFID-Tags wieder aus dem Kreis der bekannten Tags und somit aus dem Kreis der getrackten Tags zu entfernen (siehe Abb. 3.1.3g).

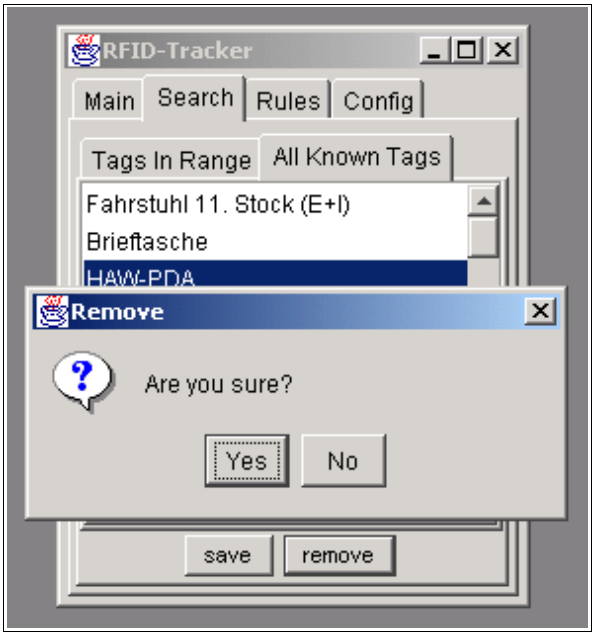

Abbildung 3.1.3g: Remove Popup-Fenster

Das Rules-Panel

Auf der Registerkarte „Rules“ soll eine Aufstellung aller angelegten Regeln erfolgen (siehe Abb. 3.1.3h).

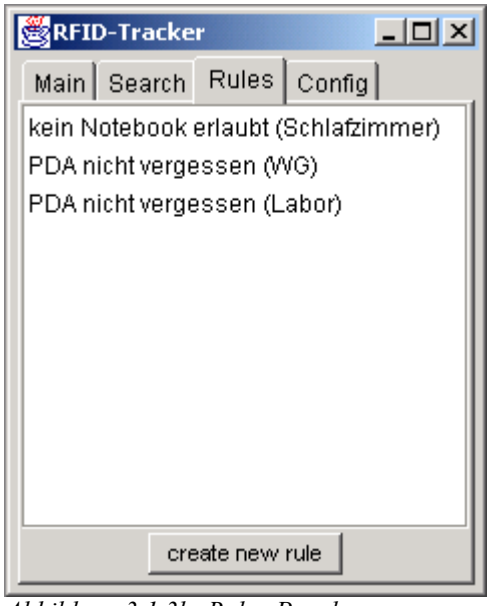

Abbildung 3.1.3h: Rules-Panel

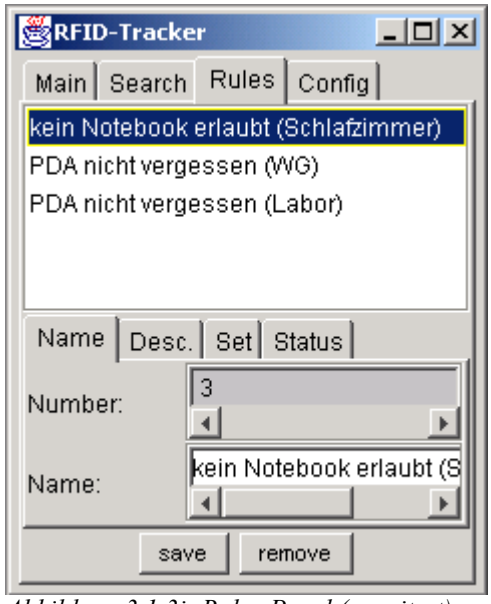

Abbildung 3.1.3i: Rules-Panel (erweitert)

Vergleichbar mit dem „Search-Panel“ kann auch hier durch das Anklicken eines Regelnamens ein zusätzliches „Konfigurationsmenü“ geöffnet werden (siehe Abb. 3.1.3i). Durch das Drücken des Knopfes „create new rule“ soll eine neue Regel angelegt werden können (siehe Abb. 3.1.3h).

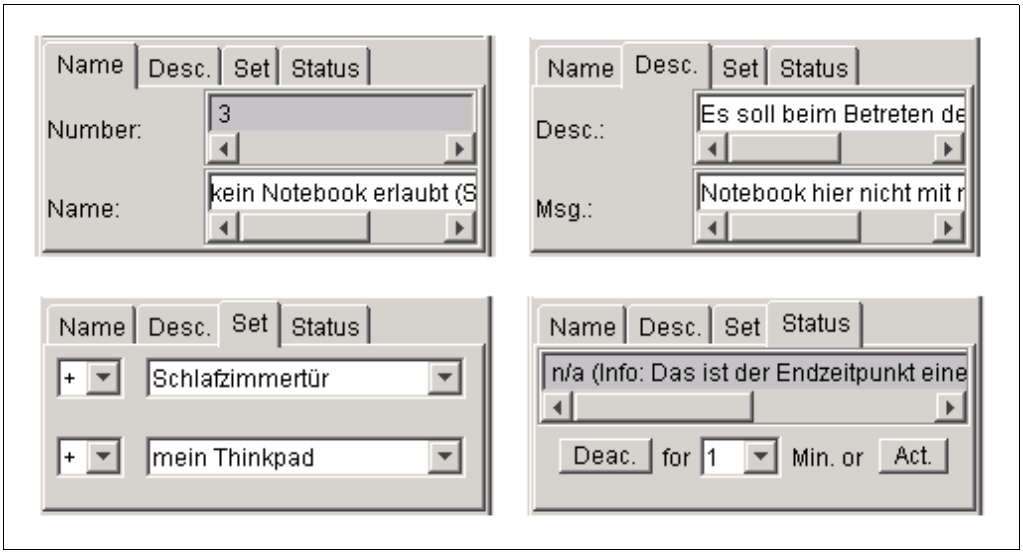

Abbildung 3.1.3j: Das Rules-"Konfigurationsmenü" im Einzelnen

Das „Konfigurationsmenü“ der einzelnen Regeln soll unter den Registerkarten „Name“, „Desc.“, „Set“ und „Status“ Möglichkeiten zum Ändern der Regeldaten anbieten (siehe Abb. 3.1.3j).

Unter „Name“ soll sich die vom System vergebene Regelnummer und der vom Benutzer bestimmbare Regelname finden lassen. „Desc.“ soll Zugriff auf den beschreibenden Text und die Fehlermeldungsnachricht geben. Unter „Set“ findet sich dann die Einstellung dafür, wie zwei RFID-Tags gleichzeitig aufzutreten haben, wenn die Regel einen Alarm auslösen soll. Unter „Status“ lässt sich der Suspendierungszustand ermitteln und ändern.

Mit dem Knopf „save“ soll sich eine editierte Regel abspeichern und mit dem Knopf „remove“ komplett entfernen lassen (siehe Abb. 3.1.3i).

Das Config-Panel

Die Registerkarte „Config“ soll es dem Benutzer ermöglichen, die Kommunikationsschnittstelle zur RFID-Watch-Box zu bestimmen (siehe Abb. 3.1.3k).

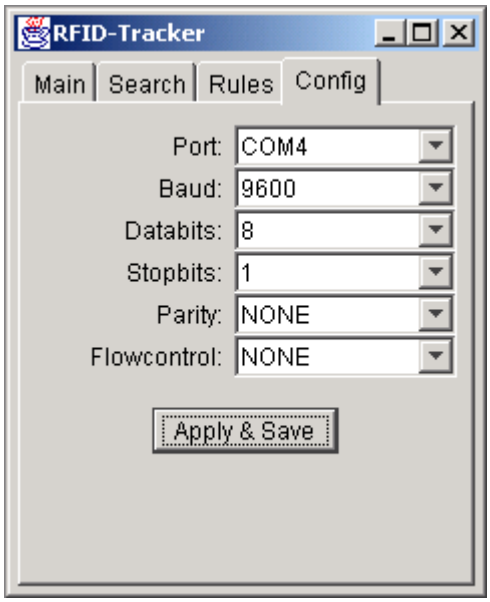

Abbildung 3.1.3k: Config-Panel

Die Combobox zur Port-Einstellung soll nur tatsächlich im System vorhandene Kommunikationsschnittstellen aufführen. Auch sollen diese automatisch so benannt werden, wie sie unter dem jeweiligen Betriebssystem heißen - zum Beispiel „COM1“ unter WindowsXP, „COM1:“ unter Windows Mobile 2003 und „/dev/ttyS0“ unter Unix oder Linux.

3.2 Analyse der „RFID-Watch-Box“

3.2.1 Festlegung der Anforderungen

Durchgängige Überwachung der Regeln

Alle Regeln, die durch den „RFID-Configurator“ aufgestellt wurden, müssen selbstverständlich auch durchgängig auf ihre Einhaltung hin überprüft werden. Durchgängig heißt in diesem Fall zu jeder Zeit und an jedem Ort. Durchgängig bedeutet auch, dass die Umgebung unentwegt nach RFID-Tags abgesucht werden muss, da schon wenige Sekunden ausreichen, um in den aktiven Bereich eines RFID-Tags wie zum Beispiel einer Landmarke hinein- und auch gleich wieder herauszutreten.

Reichweite des RFID-Lesegeräts

Wie auf dem Bild 3.2.1a zu sehen ist, muss bei der Auswahl des Lesegeräts und dessen Reichweite, auch bedacht werden, dass alle Landmarken, die man passieren möchte, garantiert im Empfangsbereich liegen.

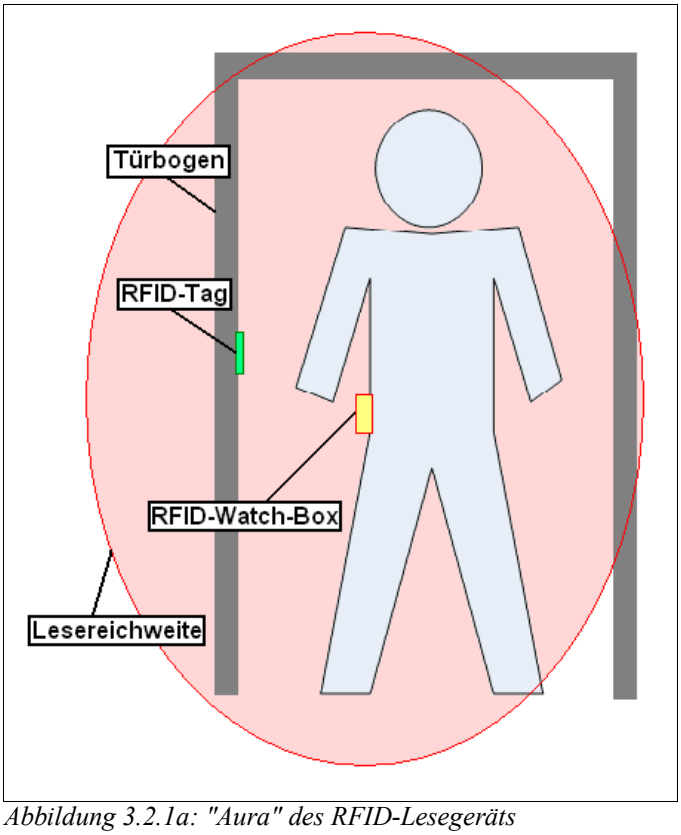

Abbildung 3.2.1a: "Aura" des RFID-Lesegeräts

Im Rahmen dieser soeben gemachten Betrachtung erscheint es ratsam das RFID-Lesegerät für eine Reichweite von mindestens ein bis besser zwei Metern auszulegen.

Alarmierung bei Regelverstößen

Sollte ein Regelverstoß erkannt werden, soll der Benutzer akustisch darauf hingewiesen werden. Der akustische Alarm soll so lange andauern, bis der Benutzer über den „RFID-Configurator“ bestätigt, dass die Regel temporär ausgesetzt werden soll. Solange ein solches Aussetzen der Regel noch nicht eingeleitet wurde, soll der bereits angefangene akustische Alarm weiter anhalten, auch wenn der Regelverstoß nicht mehr aktuell ist.

Ein Beispiel dafür: Das System meldet über einen akustischen Alarm, dass der RFID-Tag, welcher die Brieftasche repräsentiert, nicht mehr in Reichweite ist. Sollte die Brieftasche nun wieder in die Reichweite des RFID-Lesegerätes kommen, soll der Alarm trotzdem weiter gehen, bis der Benutzer über den „RFID-Configurator“ bestätigt, dass dieser Regelverstoß in Ordnung war bzw. von ihm wahrgenommen wurde, indem er die Regel für eine oder mehr Minuten suspendiert.

Regelverstöße temporär zulassen

Es kann natürlich vorkommen, dass man schon vor einem Regelverstoß weiß, dass man wohl wissentlich temporär gegen eine aufgestellte Regel verstoßen möchte. Da man in so einem Fall jedoch bestimmt nicht ständig auf das Eintreten dieser gewollten Regelverstöße hingewiesen werden möchte, muss dass System dem Benutzer auch bereits vor dem Auftreten eines Alarms die Möglichkeit bieten, einzelne Regeln zeitlich befristet zu suspendieren. Diese Forderung nach einer Möglichkeit für die Suspendierung von Regeln wurde auch schon in der Festlegung der Anforderungen für den „RFID-Configurator“ angesprochen.

Lange Akkulaufzeiten und harte Realzeit

Wie schon erwähnt, sollen die Regeln durchgängig überwacht werden. Um dieses gewährleisten zu können, muss das Gerät Akkulaufzeiten von mindestens 10 Stunden aufweisen, um mindestens einen Arbeitstag lang ununterbrochen funktionieren zu können. Zudem kommt noch die Forderung, dass die Umgebung mindestens im Sekundentakt nach RFID-Tags abgesucht werden muss, damit keine möglichen RFID-Tag-Sichtungen durch zu schnelles vorbeigehen verloren gehen. Dieser Suchrhythmus, der bereits eine harte Realzeitbedingung darstellt und die Forderung nach langen Akkulaufzeiten, legen den Wunsch nahe, ein dediziertes Gerät mit dieser Aufgabe zu betrauen. Dieses dedizierte Gerät soll ausschließlich dazu dienen, die Umgebung nach RFID-Tags abzusuchen und die Regeleinhaltung zu überprüfen. Durch unsere schon gemachte Festlegung darauf, dass wir

die Systemkomponente „RFID-Watch-Box“ durch ein embedded System realisieren möchten, ist die Forderung nach einem dedizierten Gerät für dieses Vorhaben bereits erfüllt.

Datenspeicherung

Um den Umfange der zu speichernden Daten in der „RFID-Watch-Box“ möglichst klein zu halten, sollen dort nur die unbedingt nötigsten Informationen zu den zu trackenden RFID-Tags und den Regeln gesichert werden. So ist zum Beispiel nur die zu trackende ID-Nummer in der „RFID-Watch-Box“ zu speichern, nicht aber der vom Benutzer vergebene Name und auch nicht der weiterführende beschreibende Text. Diese Zusatzinformationen, die dem Benutzer einen einfacheren Umgang mit dem System ermöglichen sollen, sind ausschließlich im „RFID-Configurator“ zu verwalten.

Kommunikation zwischen dem Akteur „RFID-Configurator“ und der „RFID-Watch-Box“

Um dem Anwender die geforderten Eigenschaften, wie sie im Kapitel 3.1 „Analyse des RFID-Configurators“ beschrieben sind, zu bieten, muss der „RFID-Configurator“ die Einstellungen in der „RFID-Watch-Box“ ändern und abfragen können. Zudem muss auch die Abfrage der gesammelten Informationen, die beim Sichten der RFID-Tags anfallen, möglich sein. Nicht zu vergessen ist auch das Einholen von Angaben über aktuell anliegende Regelverletzungen. Jede Kommunikation zwischen „RFID-Configurator“ und der „RFID-Watch-Box“ soll erstmal vom „RFID-Configurator“ als abfragender Part ausgehen.

Es ist wünschenswert, dass die Möglichkeit besteht, dass in einer späteren Erweiterung der „RFID-Configurator“ der „RFID-Watch-Box“ mitteilen kann, dass er von nun an automatisch über alle Veränderungen der in Reichweite befindlichen RFID-Tags informiert werden möchte. So würde die „RFID-Watch-Box“ alsdann Veränderungsmeldungen ohne Einzelaufforderung aussenden, sodass das ständige Pollen durch den „RFID-Configurator“ überflüssig werden würde. Für den hier zu bauenden Prototyp soll aber erstmal nur die pollende Variante berücksichtigt werden.

3.2.2 Ermittlung der Anwendungsfälle

Nun folgend werden die Anwendungsfälle, in denen der „RFID-Configurator“ den Part des Akteuren übernimmt, in Form von Use-Case-Diagrammen dargestellt.

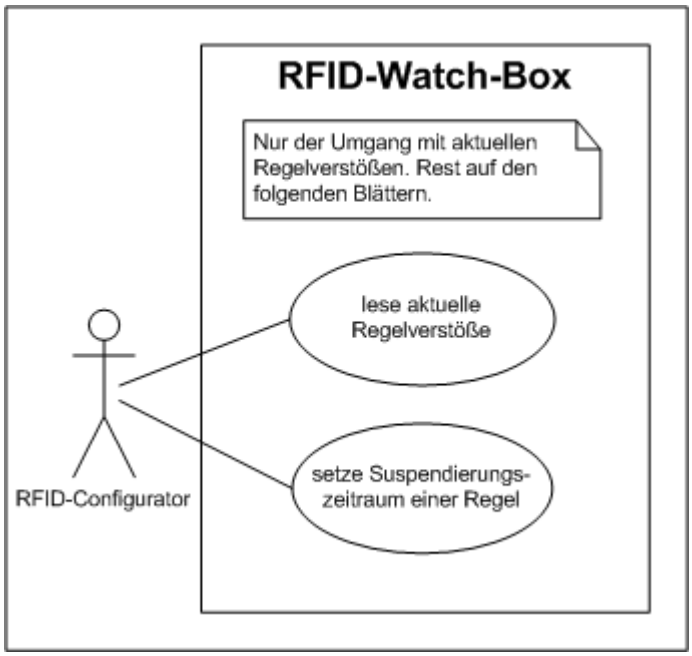

Abbildung 3.2.2a: Use-Case-Diagramm “Meldung von und Umgang mit aktuellen Regelverstößen“

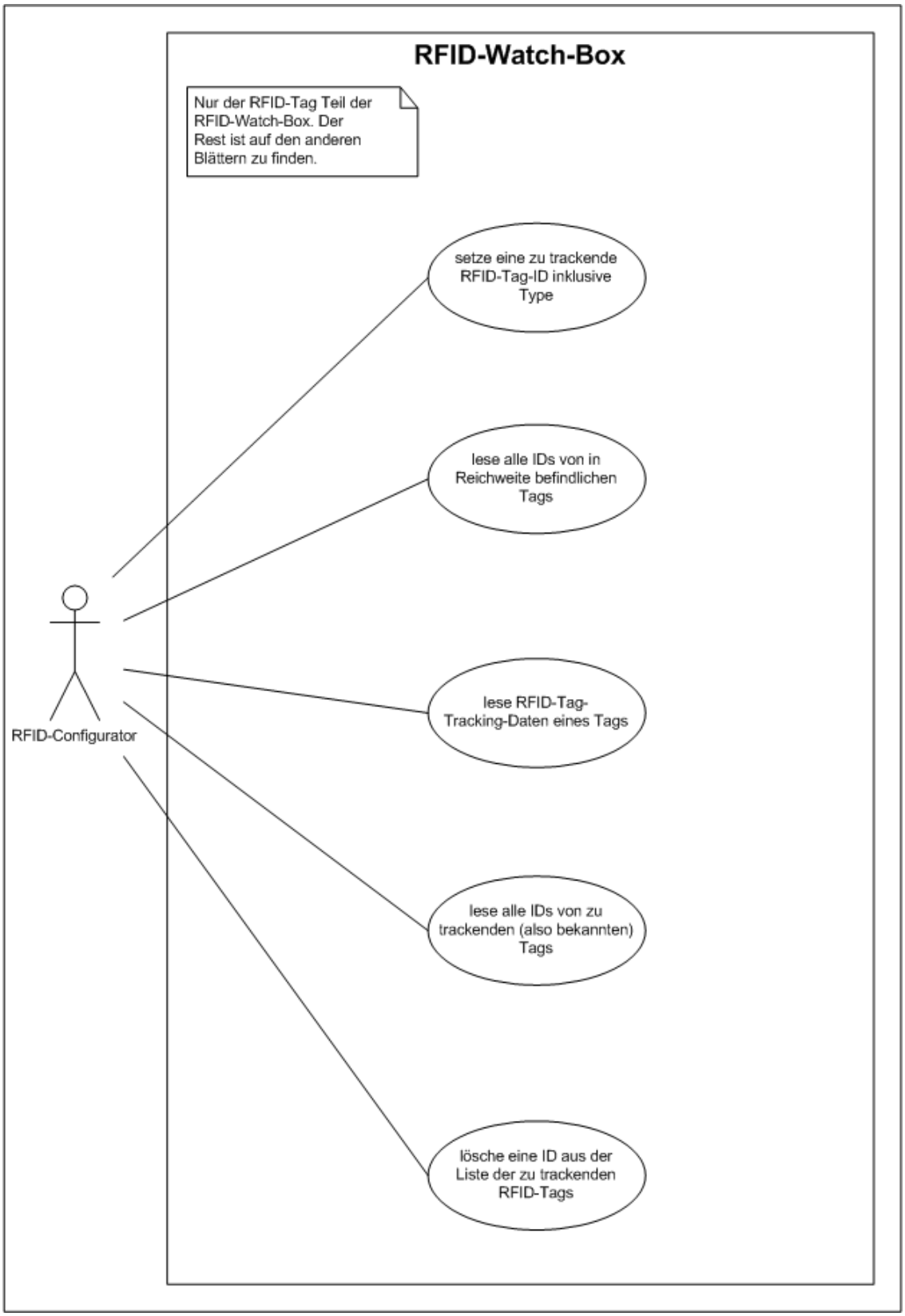

Abbildung 3.2.2b: Use-Case-Diagramm "Verwaltung von RFID-Tags"

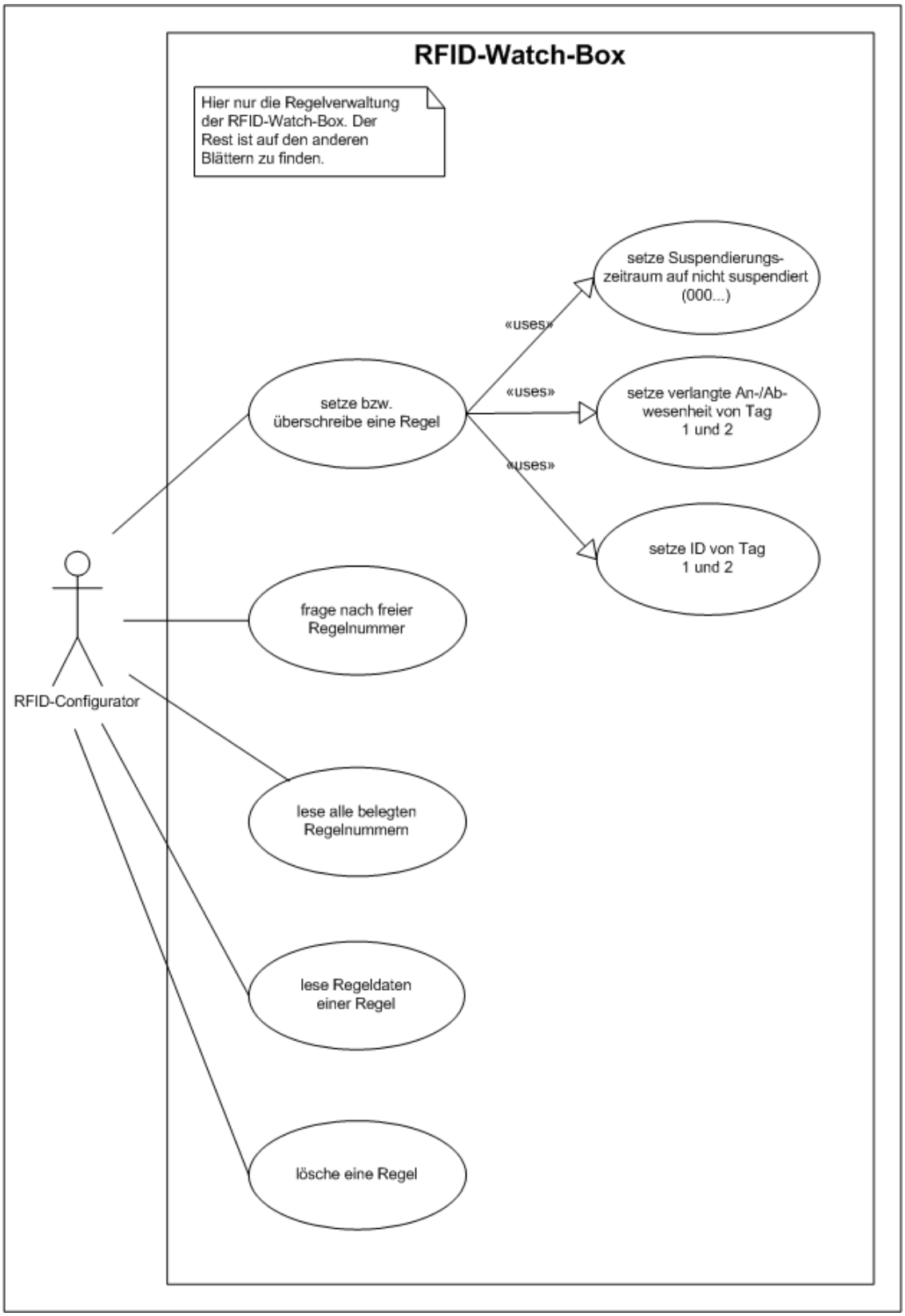

Abbildung 3.2.2.c: Use-Case-Diagramm "Verwaltung von Regeln"

3.2.3 Zusätze

Zusätzlich zu den in den Use-Case-Diagrammen dargestellten Anwendungsfällen werden noch die Anwendungsfälle „setze Uhrzeit und Datum“ und „lese Uhrzeit und Datum“ benötigt. Sie dienen dazu, die aktuelle Systemzeit der „RFID-Watch-Box“ zu setzen beziehungsweise auszulesen. Zu guter Letzt soll noch der Anwendungsfall „frage nach, ob im Moment eine Anfrage von der RFID-Watch-Box entgegengenommen werden kann“ dazu dienen, dass der „RFID-Configurator“ bei der „RFID-Watch-Box“ anfragen kann, ob im Moment überhaupt eine Nachricht beziehungsweise ein Befehl abgearbeitet werden kann. Dadurch soll überflüssiger Datenverkehr zwischen den beiden vermieden werden.

3.3 Exkurs: Sicherheitsanalyse

Ein gewisses Problem, welches eine groß angelegte Einführung von vielen der durch Wirtschaft und Politik geplanten RFID-Systemen behindern könnte, stellt die allgemeine Furcht der Bevölkerung vor unvorhersehbaren Eingriffen in ihre Privatsphäre dar. Hatte in den frühen achtziger Jahren des vergangenen Jahrhunderts die Volkszählung noch große Empörung verursacht, so haben sich die Bürger mittlerweile sogar schon an solche Dinge, wie die stetige Videoüberwachung auf Bahnhöfen und in U-Bahnen gewöhnt - ja sogar an die Preisgabe ihrer Konsumgewohnheiten durch personenbezogene Rabattkartensysteme. Selbst eine groß angelegte Gen-Datei zur effizienteren Strafverfolgung ist seit neuestem wieder ernsthaft im Gespräch. Aber dennoch könnte womöglich das eine oder andere RFID-Projekt die Toleranz der Bevölkerung überspannen.

Auf alle Fälle lohnt es sich, sich auch mal mit Sicherheitsaspekten dieser Technik zu beschäftigen. Vor allem im Sinne der Informationsvertraulichkeit, da damit wohl auch das Vertrauen der Bevölkerung in diese relativ junge Technik stehen oder fallen wird. Hierin liegt dann auch der primäre Grund für diesen Exkurs. Es sollen jedoch auch Aspekte beleuchtet werden, die zum Beispiel die Frage aufkommen lassen werden, ob sich RFID-Systeme überhaupt als alleiniges Mittel zur sicheren Authentifizierung von Personen eignen oder nicht - zumal es bereits eine Vielzahl von Anlagen gibt, bei denen man darauf vertraut.

Die Gefahren die durch eine groß angelegt Sammlung und Weiterverarbeitung von Daten aus RFID-Systemen entstehen, sind schwer bis vielleicht gar nicht abzuschätzen. Selbst wenn nur berechtigte Betreiber und Nutzer Zugriff auf diese gesammelten Daten haben. Diese Problematik soll hier dann auch gar nicht erst behandelt werden. Wer sich dennoch gerne mal mit diesem Thema auseinander setzen möchte, sei hier auf das Paper „Privacy Enhancing Technology Concepts for RFID Technology Scrutinised“ [25] der RVS[1]-Gruppe an der Universität Bielefeld verwiesen. Wer sich hingegen über die üblichen und meist von außen kommenden Angriffsmöglichkeiten wie zum Beispiel DoS[2]-Attacken speziell bezogen auf RFID-Systeme informieren möchte, sei ein Blick in die RIKCHA[3]-Studie [26] des BSI[4] empfohlen.

Hier werden nun folgend zwar auch einzelne Angriffe beschrieben, wie sie auch in anderen Computersystemen vorkommen können und dort auch allgemein bekannt sind, vor allem soll es aber darum gehen, tiefer liegende Angriffspunkte anzusprechen, unter denen auch eine Vielzahl anderer Systeme abseits der RFID-Welt leiden, ohne dass jedoch darauf im Normalfall geachtet wird.

1 Rechnernetze und verteilte Systeme
2 Denial of Service
3 „Risiken und Chancen des Einsatzes von RFID-Systemen“
4 Bundesamt für Sicherheit in der Informationstechnik

3.3.1 Fehlende Anonymität

Eine grundlegende Eigenschaft der RFID-Tags ist, dass sie tatsächlich eindeutige IDs aufweisen. Hier sei zur Verdeutlichung ein Beispiel aus dem Einzelhandel angeführt: Die bis heute eingesetzten Strichcodes auf den Waren beinhalten eine so genannte EAN-Nummer. Diese 13-stellige EAN-Nummer ist von Produkt zu Produkt unterschiedlich, so dass man ein Tetrapack Milch problemlos von einer Flasche Orangensaft unterscheiden kann. Will man jedoch zwei Tetrapacks der gleichen Milchsorte voneinander unterscheiden, so hilft einem die EAN-Nummer auch nicht weiter. Selbst wenn diese beiden Tetrapacks zu gänzlich unterschiedlichen Zeitpunkten abgefüllt worden sind und bei einer schon längst das Haltbarkeitsdatum überschritten ist, ist eine Unterscheidung anhand der EAN nicht machbar. Erst RFID-Tags sollen es ermöglichen jedes einzelne Erzeugnis eindeutig zu identifizieren. Eindeutige Seriennummern anstatt generische Produktbezeichnung ist das Zauberwort. Hierzu senden RFID-Tags ihre eindeutige ID aus, sobald sie in den Bereich eines RFID-Lesegeräts kommen. Diese ID ist selbstverständlich viel länger als die 13-stellige EAN. Diese ausgesendete ID stellt zudem auch die Adresse des RFID-Tags dar, über die es von einem Lesegerät direkt angesprochen werden kann, um z.B. Lesebefehle auf einem in dem RFID-Tag gegebenenfalls vorhandenen Datenspeicher auszuführen.

Ein Lösungsansatz, um das unberechtigte Abfragen der eindeutigen ID zu verhindern, wäre die Einführung eines dynamischen Adressraumes. Ein RFID-Tag könnte sich dann, wenn es in den Bereich eines RFID-Lesegeräts kommt, statt seiner ID eine Adresse aus dem dynamischen Adressraum entleihen und diese als seine aktuelle Adresse aussenden. So könnten dann zum Beispiel die nächsten 20 adressierten Befehle mit dieser dynamischen Adresse als Ziel akzeptiert werden. Nach diesen 20 adressierten Befehlen oder nach einer Kollision mit einem RFID-Tag mit der selben dynamischen Adresse oder nach Betreten eines Lesebereichs eines anderen RFID-Lesegeräts könnte ein Pseudozufallsgenerator[1] eine neue Adresse generieren, die dann wiederum ausgesendet wird. Innerhalb der vorgeschlagenen 20 adressierten Befehle hat sich das Lesegerät mit Zuhilfenahme von kryptographischen Verfahren gegenüber dem RFID-Tag zur Abfrage der tatsächlichen ID zu autorisieren, um dann auch tatsächlich die eindeutige ID in Erfahrung zu bringen.

Eine interessante Quelle für weitere Informationen zu diesem Thema stellt das Paper „Die Privatsphäre im Ubiquitous Computing - Datenschutzaspekte der RFID-Technologie“ [27] dar.

1 Hierbei ist natürlich darauf zu achten, dass auch die Reihenfolge der gewählten dynamischen Adressen keine Identifizierung eines einzelnen RFID-Tags zulässt.

3.3.2 „Feind hört mit“

Bedingt durch die relativ hohen Sendeleitung der Lesegeräte, ist die Kommunikation von Lesegerät zu RFID-Tag im Normalfall relativ leicht abzuhören. Aber auch die Kommunikation von Tag zu Lesegerät darf nicht als abhörsicher angesehen werden. Selbst bei passiven Tags, welche über Lastmodulation ihre Antwort an das Lesegerät übermitteln, kann ein Abhören nicht ausgeschlossen werden [49]. Spätestens Gerätschaften, wie sie in Kapitel 3.3.3 beschrieben werden, eröffnen einem den Weg beide Kommunikationswege problemlos zu überwachen. Den einzig sicheren Ansatz kann hier nur die Kryptographie liefern, indem man auf die Verschlüsselung der Kommunikation setzt.

3.3.3 Der Repeater-Angriff

Sehr beliebt sind RFID-Systeme mittlerweile als Zutrittssicherungsverfahren geworden [31] [32][33]. Hierbei trägt die Person, welche dazu berechtigt ist, sich Zugang zu einem bestimmten Gebäudebereich zu verschaffen, einen RFID-Tag bei sich, welcher zum Öffnen der Tür oder Durchgangsschleuse in die Nähe eines dort aufgestellten RFID-Lesegerätes gehalten werden muss. Der Benutzer authentifiziert sich also durch den Besitz des Tags. Der Vorteil gegenüber herkömmlichen Smartcard-Systemen liegt darin, dass durch das Fehlen von ständig wiederkehrenden, mechanischen Kontaktiervorgängen, ein Ausfall durch auf diese Weise entstandenen Verschleiß ausgeschlossen ist. Auch ist die Anfälligkeit gegenüber Vandalismus stark reduziert (siehe auch Kapitel 2.1). Je nach Qualität der RFID-Systeme handelt es sich mitunter nur um sehr einfache RFID-Tags, die ausschließlich ihre ID aussenden, so dass das System lediglich anhand der empfangenden ID entscheiden muss, ob die Tür geöffnet werden darf oder nicht. Es gibt aber auch bereits kryptographisch arbeitende Systeme, bei denen die Identität von Lesegerät und Tag gegenseitig authentifiziert wird. So sind bei diesen höherwertigen Systemen „kopierte“ RFID-Tags mit der selber ID, wie die des Berechtigten, nicht mehr ausreichend, um das System zu überlisten.

Jedoch könnte auch bei den kyptographischen Systemen dadurch ein Problem entstehen, dass Lesegerät und RFID-Tag zur Kommunikation nicht direkt elektrisch miteinander verbunden sind, sondern dass mehrere Zentimeter Luft zwischen ihnen liegen. Wenn es jetzt möglich ist, aus den wenigen Zentimetern mehrere Meter, wenn nicht sogar Kilometer zu machen, kann es sein, dass sich vor der zu öffnenden Tür eine ganz andere Person befindet als diejenige, welche dazu berechtigt ist, die Tür zu öffnen. Hierbei würde somit die berechtigte Person weiterhin wie gewohnt den passenden RFID-Tag bei sich tragen und hätte quasi keine Chance zu merken, dass seine Zugangskarte von jemand anderem quasi mitbenutzt wird.

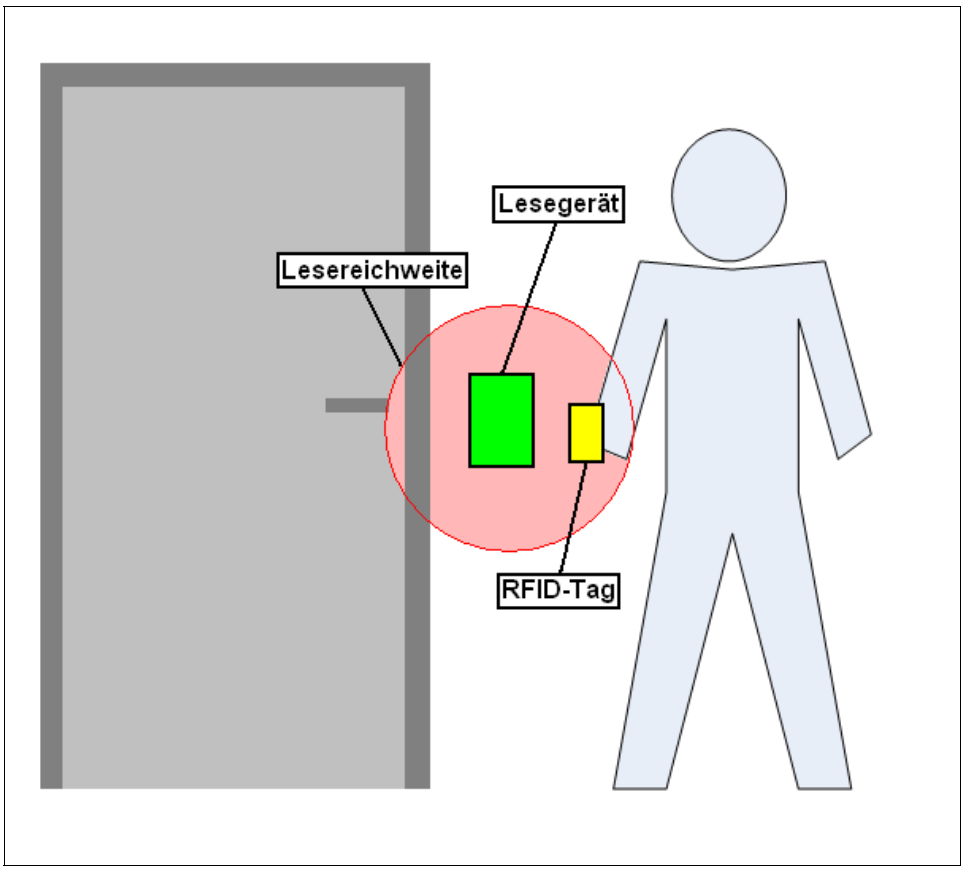

Abbildung 3.3.3a: Normale Reichweite

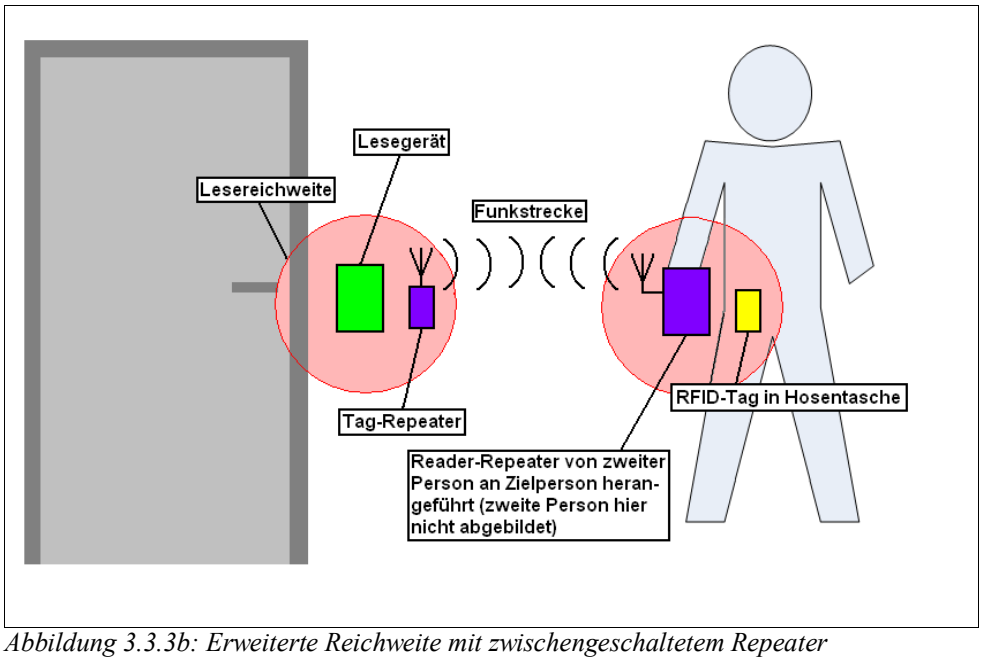

Abbildung 3.3.3b: Erweiterte Reichweite mit zwischengeschaltetem Repeater

Doch wie soll eine solche Reichweitenerhöhung vonstatten gehen? Bild 3.3.3b zeigt uns eine Übersicht über das Angriffskonzept. Auf der linken Seite sehen wir die gesicherte Tür mit dem RFID-Lesegerät. Anstatt des richtigen RFID-Tags wird ein Tag-Repeater vor das Lesegerät gehalten. Dieser Tag-Repeater verhält sich gegenüber dem Lesegerät genau so wie ein RFID-Tag, nur dass er Anfragen des Lesegeräts nicht direkt selber beantwortet, sondern an den Reader-Repeater bit- bzw. taktsynchron weiterleitet. Hierzu sollen der Tag-Repeater und der Reader-Repeater zum Beispiel durch eine FSK-modulierte Duplex-Funkverbindung im 70cm Bereich miteinander verbunden sein. Wenn nun die Anfrage des Lesegeräts Bit für Bit beim Reader-Repeater ankommt, tut dieser gegenüber dem RFID-Tag so, als ob er ein RFID-Lesegerät sei und sendet genau die gestellte Anfrage des echten RFID-Lesegeräts an den RFID-Tag. Der RFID-Tag antwortet alsdann auf diese Anfrage, welche der Reader-Repeater empfängt und ebenfalls bit- bzw. taktsynchron an den Tag-Repeater weiterreicht. Der Tag-Repeater antwortet dem RFID-Lesegerät nun so, als ob er das zum Zugang berechtigende RFID-Tag sein würde. Es ist vollkommen egal, ob RFID-Lesegerät und RFID-Tag ihre Kommunikation in irgendeiner Weise verschlüsseln oder ob sie sich gegenseitig ihre Identitäten kryptographisch authentifizieren, da die gesamte Kommunikation einfach nur Bitweise weitergereicht wird. Ein Lösungsansatz gegen dieses Problem wäre, die Signallaufzeiten bzw. Antwortzeiten zu messen. Es besteht zur Zeit sogar eine Diplomarbeitsausschreibung des Frauenhofer IPA, welche zum Thema hat, sich damit zu beschäftigten, aus den Signallaufzeiten die Entfernung zwischen RFID-Lesegerät und RFID-Tag zu bestimmen [28]. Leider reicht ein kurzer Blick in die Spezifikation von vielen passiven RFID-Tags, dass diese ihren Systemtakt aus der Sendefrequenz des RFID-Lesegeräts ableiten. Wenn man nun noch bedenkt, dass RFID-Tags so ausgelegt sind, dass sie sehr tolerant gegenüber Frequenzabweichungen des Lesegeräts sind, um Frequenzbeeinflussung durch Umgebungseinflüsse und Produktionstoleranzen entgegenzuwirken, kommt man sehr schnell zu dem Schluss, dass man einfach nur die Frequenz des Reader-Repeaters um einige Kilohertz gegenüber der Sendefrequenz des echten RFID-Lesegeräts erhöhen muss, um die zusätzliche Laufzeitverzögerung wieder auszugleichen. Eine Lösung für das Problem des Repeater-Angriffs könnte in einer HF-dichten Hülle liegen, in die der RFID-Tag geschoben wird, wenn man nicht gerade im Begriff ist, eine Tür mit ihr öffnen zu wollen. Systeme wie der VeriChip, welcher unter die Haut der zugangsberechtigten Person implantiert wird, wie er bei der mexikanischen Staatsanwaltschaft eingesetzt wird [29], könnte zwar durch eine umgelegte HF-dichte Manschette abgeschirmt werden, aber in wieweit dieses Vorgehen in der Praxis praktikabel ist, dürfte zu bezweifeln sein. Sollten RFID-Tags im EC-Karten-Format eingesetzt werden, kann man auch über die Verwendung von kapazitiv gekoppelten Tags (close-coupling) nachdenken, da hier wohl bereits die normale Brieftasche als HF-dicht angesehen werden kann. Aber die praxistauglichste Lösung für Zugangssysteme wird wohl sein, RFID-Tag-Karten mit einem Taster zu versehen, welcher dafür sorgt, dass der RFID-Tag nur antworten kann, wenn der Taster gedrückt wird.

Man sollte ebenfalls bedenken, dass auch in anderen RFID-Anwendungen ein solcher Angriff schaden hervorrufen kann. So wäre es zum Beispiel vorstellbar, dass im ÖPNV

mehrere Personen mit ein und derselben RFID-Monatskarte fahren. Selbst wenn die Uhrzeiten der genauen Ein- und Ausstiege erfasst und on- oder offline abgeglichen werden sollten, könnten mit der beschriebenen Methode immer noch mehrere Personen sich eine Karte teilen. Sie müssten sich nur dahingehend absprechen, nicht gleichzeitig den ÖPNV zu nutzen, ansonsten bleiben sie jedoch unabhängig voneinander und könnten sich getrennt voneinander im ÖPNV-Bereich bewegen. Viele weitere Szenarien dieser Art sind vorstellbar, so dass jede RFID-Anwendung, in dieser Hinsicht gründlich überprüfen werden sollte. Auch Ansätze, wie Straftäter durch RFID zu überwachen, erscheinen in diesem Blickwinkel äußerst problematisch [30].

3.3.4 Der „Funk-Fingerabdruck“

Der „Funk-Fingerabdruck“ ist eine Technik, die beispielsweise von staatlichen Funkmessdiensten eingesetzt wird, um unbekannte und zumeist nicht ortsfeste Sendeanlagen wiedererkennen zu können. Hierbei macht man sich zu Nutze, dass jede Hochfrequenzsendeanlage, welche eine definierte Trägerfrequenz verwendet, beim Hochtasten[1] ein spezifisches Einschwingverhalten an den Tag legt[2]. Dieser „elektronische Fingerabdruck“ beziehungsweise „Funk-Fingerabdruck“ auf Seiten des RFID-Lesegeräts sollte bei den meisten RFID-Systemen irrelevant sein, da dort die Lesegeräte ortsfest sind und ein unberechtigtes Wiedererkennen des Lesegeräts wohl keinen Wert darstellen sollte. Anders sieht es hingegen bei dem hier zu entwickelnden RFID-Tracker aus. Hier kann das Lesegerät einer Person zugeordnet werden, da diese das Lesegerät ständig bei sich trägt. Ein Wiedererkennen des Lesegeräts lässt also auch Rückschlüsse auf den aktuellen Aufenthaltsort des Besitzers zu.

Im Kontext des „Funk-Fingerabdrucks“ wäre es sicherlich auch interessant, mal zu untersuchen, in wieweit ebenfalls die RFID-Tags charakteristische Signalverläufe innerhalb ihrer Modulation aufweisen.

In Bezug zu diesem Thema sollte der interessierte Leser einen Blick in die US-Patente 5.758.277 [34] und 5.005.210 [35] werfen (auch im Anhang). Die Firma Motron bietet auf ihrer Homepage ein komplettes Soft- und Hardwarepaket für das „Transmitter Fingerprinting“ an [36]. Zudem kann man im Internet auch ein nicht kommerzielles Softwareprojekt finden, welches sich mit diesem Thema beschäftigt [37].

1 Beginn der Sendeaktivität
2 Dass es auch charakteristische Amplitudenschwankungen während des Aussendung und kennzeichnende Frequenzschwankungen am Ende einer Aussendung gibt, soll hier nur am Rande erwähnt sein, schon da zumindest die Auswertung der Amplitudenschwankungen sehr fehleranfällig ist.

3.3.5 Funkpeilung

Ein RFID-Lesegerät, welches, wie es unter „der Funk-Fingerabdruck“ beschrieben wurde, wiedererkannt werden konnte, könnte auch per Funkpeilung über weite Strecken von dazu nicht autorisierten Personen in seiner Bewegung weiterverfolgt werden. Die verwendeten Sendeleistungen der Lesegeräte zwischen 100 mW und 2 Watt sind zwar nur ausreichend, um die RFID-Tags wenige Zentimeter bis Meter weit auszulesen, für einem Funkempfänger mit Peileinrichtung reichen aber in schwach bebauten Gebieten schon die 100 mW, um noch aus über 1000 Metern ein verwertbares Signal problemlos zu empfangen.

Dazu könnten manuell betriebene TDOA-Antennen (Time-Difference-Of-Arrival) zum Einsatz kommen oder aber auch voll automatisch arbeitende Doppler-Peiler, welche an verschiedenen Orten aufgestellt, die genaue Position der HF-Aussendung vollautomatisch per Triangulation bestimmen könnten.

TDOA-Antennen[1]

Die preiswerten TDOA-Antennen finden zum Beispiel Anwendung in der Tierverhaltensforschung in freier Wildbahn. Hierbei wird den Tieren von den Biologen ein kleiner Sender angelegt, welcher in regelmäßigen Abständen auf einer bestimmten Frequenz kurz einen unmodulierten Träger aussendet. Die Sendeleistung, Sendehäufigkeit und Batteriekapazität wird dabei im Normalfall so ausgelegt, dass diese Peilsender sechs Monate und länger aktiv bleiben können.

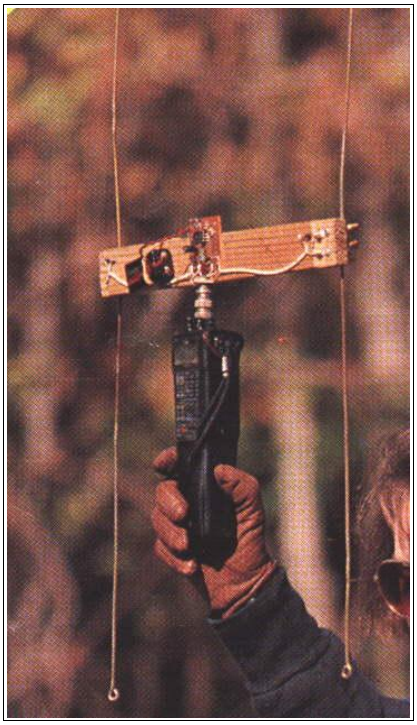

Abbildung 3.3.5a: TDOA-Antenne

Die TDOA-Antenne am Peilempfänger besteht hierbei eigentlich aus zwei einzelnen Antennen, zwischen denen schnell und regelmäßig hin- und hergeschaltet wird. Der Empfänger ist also abwechselnd mit der einen oder mit der anderen Antenne verbunden. Wenn nun die beiden Antennen unterschiedlich weit vom angepeilten Sender entfernt stehen, ergibt sich ein Phasenunterschied zwischen den beiden empfangenen Signalen. Dieser Phasensprung beim Umschalten zwischen den Antennen ist im Empfänger als Klack-Geräusch zu hören. Wenn nun die Umschaltfrequenz zwischen den beiden Antennen bei beispielsweise 500 Hz liegt, hat auch der Ton, der durch den Phasensprung zustande kommt, eine Frequenz von 500 Hz. Wenn man die TDOA-Antenne

1 Ein Selbstbauprojekt für eine TDOA-Antenne ist dort zu finden: [38]

nun derart dreht, dass beide Einzelantennen den selben Abstand zum Peilsender haben, erlischt der 500 Hz Ton und man erkennt aus der Ausrichtung der beiden Einzelantennen zueinander, die Richtungsachse aus der das Signal kommt (siehe Abb. 3.3.5b). Nun sind aber noch zwei genau gegenüberliegende Seiten als Empfangsrichtung möglich. Um dieses Problem zu lösen, kann man an zwei Orten gleichzeitig eine Messung vornehmen und über Triangulation nicht nur die richtige Richtung, sondern sogar noch die Entfernung des Signals bestimmen (siehe Abb. 3.3.5c).

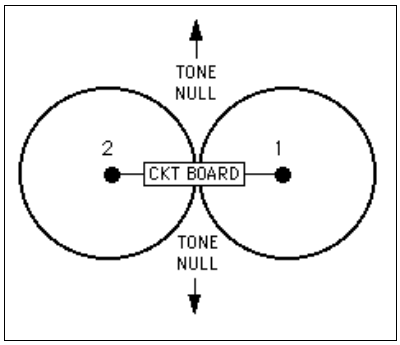

Abbildung 3.3.5b: Richtungsachse

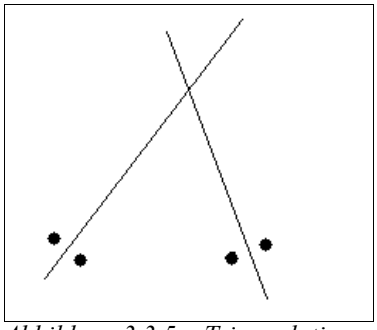
Abbildung 3.3.5c: Triangulation

Sollte es jedoch nicht möglich sein, zwei Messungen gleichzeitig vorzunehmen, kommt auch noch eine modifizierte Version der TDOA-Antennen in Frage, welche nicht nur in der Lage ist, die korrekte Richtungsachse zu bestimmt, sondern auch gleich die richtige Seite ermitteln kann. Hierzu wird zwischen einer der beiden Antennen und dem automatischen Antennenumschalter ein weiteres Kabel zur Signalverzögerung zwischengeschaltet. Dieses Signalverzögerungskabel ist ein ganz normales Antennenkabel, welches jedoch in seiner Länge genau darauf abgestimmt wird, dass die Phase des Signals an beiden Antennen genau gleich ist, wenn die beiden Antennen zum Ziel hin genau hintereinander liegen. Diese Phasenanpassung stimmt nur, wenn die unverzögerte Antenne vor der verzögerten Antenne liegt, aber nicht andersherum (siehe Abb. 3.3.5d). Mit einer derart modifizierten TDOA-Antenne kann man zwar die richtige Seite bestimmen, jedoch lässt sich die genaue Richtung mit dieser nur sehr ungenau ermitteln. Darum sind diese erweiterten TDOA-Antennen auch wieder auf „Normalfunktion“ umschaltbar, um nach der groben Richtungsmessung noch eine präzisiere Peilung machen zu können.

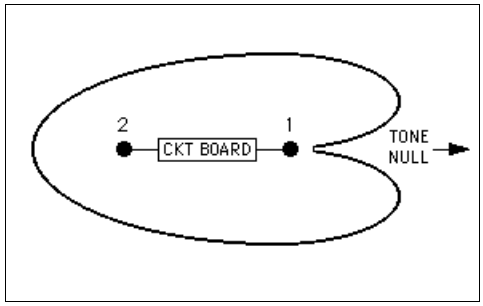

Abbildung 3.3.5d: Richtung

Doppler-Peiler[1] [2]

Dort wo eine fortlaufende und automatisierte Funkpeilung stattfinden soll, wird häufig auf das Doppler-Peilverfahren zurückgegriffen. Hierbei handelt es sich um eine Antenne, die sich am Rand einer runden Scheibe befindet, wobei sich die Scheibe mit konstanter Winkelgeschwindigkeit um die eigene Achse dreht (siehe Abb. 3.3.5e).

Da sich die Antenne auf ihrem kreisförmigen Weg immer wieder auf das Funksignal zu und auch wieder weg bewegt, kommt es hier durch den Dopplereffekt dazu, dass die empfangende Frequenz mal höher und mal niedriger ist, obwohl die abgestrahlte Frequenz die gleiche bleibt. Aus den Zeitpunkten, wann die Frequenz höher und wann sie tiefer wird, können Rückschlüsse auf die Richtung, aus der das Signal kommt, gezogen werden. Da eine sich mechanisch drehende Scheibe, auf der eine Antenne montiert ist, großem Verschleiß unterliegen würde, verwendet man stattdessen eine kreisförmig Phalanx von gleichartigen Antennen, zwischen denen in einer ebenso kreisförmigen Sequenz hin- und hergeschaltet wird (siehe Abbildung 3.3.5f).

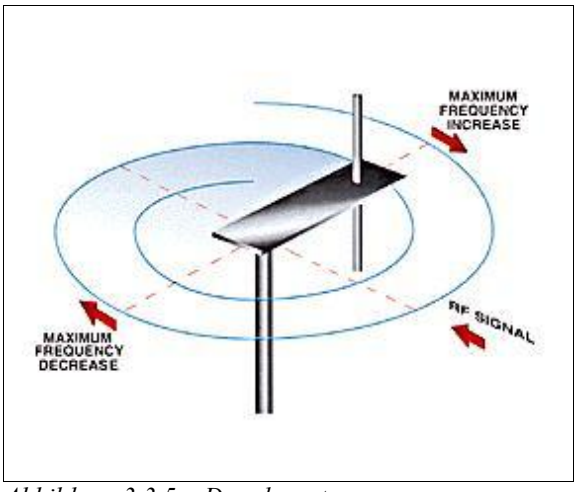

Abbildung 3.3.5e: Dopplerantenne

Abbildung 3.3.5f: Dopplerantenne

1 Dopplereffekt: Ein bekanntes Beispiel für den Dopplereffekt ist die Tonhöhenänderung des Martinshorns eines Polizeiwagens, der sich auf uns zu- beziehungsweise wegbewegt. Solange sich das Fahrzeug uns nähert, ist der Ton höher, als wenn das Fahrzeug stehen würde. Wenn es sich entfernt, ist der Ton tiefer, als wenn es stünde.

2 Im Internet sind viele Doppler-Peiler Selbstbauprojekte zu finden. Hier zwei davon: [39][40]

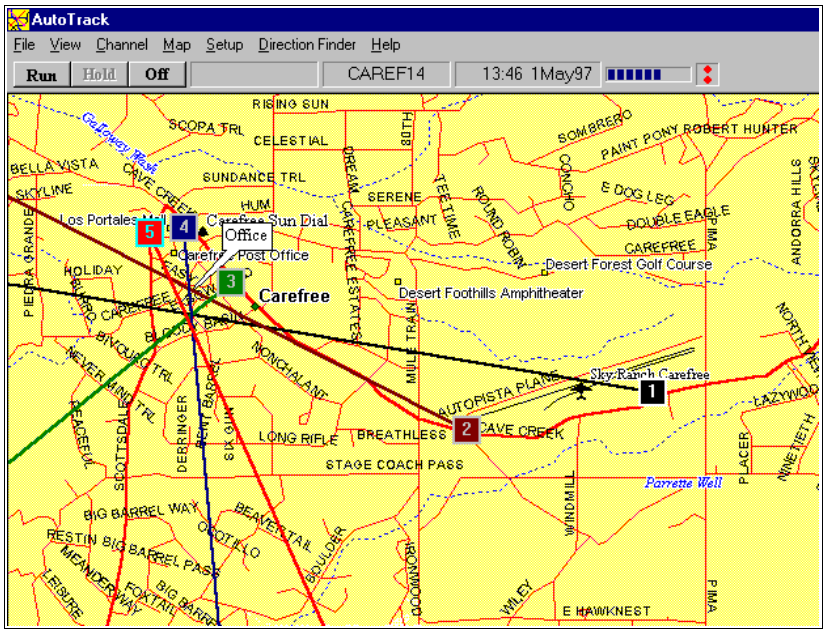

Abbildung 3.3.5g: Trackingsoftware für mobile Fahrzeugempfänger

Abbildung 3.3.5h: Professioneller Funkpeiler von Rohde&Schwarz, welcher sogar digitale Funkgeräte auf TDMA-Basis (Time Division Multiple Access), wie zum Beispiel GSM-Telefone und TETRA-Funkgeräte, anpeilen kann

Abbildung 3.3.5k: Selbstbau Doppler-Peiler

Abbildung 3.3.5i: Portabler Funkpeiler von Rohde&Schwarz

Abbildung 3.3.5j: Verdeckte Doppler-Antennen auf PKW

3.3.6 Kompromittierende Strahlung

Reine Softwaretechniker, die sich um die Sicherheit (im Sinne der Vertraulichkeit) eines Systems kümmern sollen, meinen leider allzu oft, dass dieses Problem einzig und allein durch sauber programmierte kryptographische Algorithmen zu lösen sei. Sie vergessen in dieser Hinsicht gerne, auch mal ihr Augenmerk auf die Hardware zu richten und riskieren so, dass Angreifer unbemerkt an Informationen kommen können, von denen sie bisher meinten, das sie doch nur über „sichere Kanäle“ übertragen werden.

Als Beispiel für so einen „sicheren Kanal“ in dem hier zu entwickelnden RFID-Tracker-System, sei nun die gewählte Realisierung der Bluetooth-Schnittstelle zwischen „RFID-Watch-Box“ und „RFID-Configurator“ kurz etwas näher betrachtet.

Der Bluetooth-Standard wird im Allgemeinen in kryptographischer Hinsicht als äußerst sicher eingestuft. Die in letzter Zeit immer wieder in der Fachpresse kursierenden Meldungen über Bluetooth-Sicherheitslöcher in Bezug auf verschiedene Handymodelle unterschiedlicher Hersteller, haben ihre Grundlagen in fehlerhaften Implementationen des

Bluetooth-Standards in den betroffenen Mobiltelefonen[1]. Angriffe, welche die allgemeine Sicherheit des Standards in Frage stellen würden, hat es in öffentlich zugänglichen Publikationen bisher noch nicht gegeben.

Nun aber nehmen wir mal das Realisierungsunterkapitel 4.1.2 vorweg und schauen uns an, wie ein Teil der „Management-Blackbox“ hardwareseitig aufgebaut sein soll und wie dort das Bluetooth-Interface angebunden sein wird (siehe Abbildung 3.3.6a).

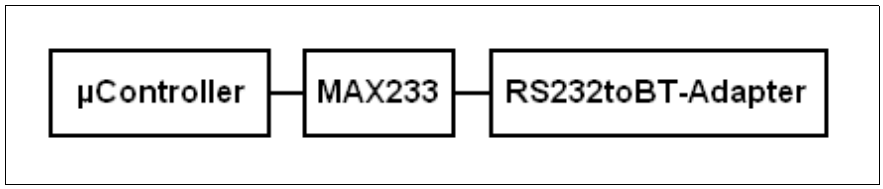

Abbildung 3.3.6a: Ein Ausschnitt aus der RFID-Watch-Box Hardware

Wir sehen, dass der µController über einen TTL zu RS232 Konverter-IC des Typs MAX233 mit dem Bluetooth-Adapter verbunden ist. In der PhD-Thesis von Markus G. Kuhn mit dem Titel „Compromising emanations: eavesdropping risks of computer displays“ [41] findet man auf der Seite 132 einen Hinweis darauf, dass der MAX233 eine 125 kHz Oszillator zum Betrieb der Ladungspumpen einsetzt. Diese Frequenz soll über die Ausgänge der RS232-Schnittstelle nach Außen gelangen können. Je nach Ausführung des daran angeschlossenen seriellen Verbindungskabels, welches nun unfreiwillig als Sendeantenne dienen kann, würde nun das Signal verschieden stark ausgesendet werden. Nach den Beobachtungen von Markus Kuhn soll sich die Oszillatorfrequenz abhängig vom aktuellen logischen Wert um einige Hertz verändern, so dass sich aus dem Empfang der Hochfrequenzstrahlung die übertragenen Daten auf der RS232-Schnittstelle rekonstruieren lassen sollen. Inwieweit dieser „Seitenausgang“ im Zusammenhang des RFID-Trackers ein tatsächliches Sicherheitsproblem darstellt, soll in diesem Rahmen erstmal nicht weiter erörtert werden. Es sollte jedoch im Hinterkopf behalten werden, dass für ein wirklich sicheres System auch solche Dinge eingehend überprüft werden sollten.

1 Eins dieser Probleme ist zum Beispiel die Möglichkeit des so genannten „Bluesnarfings“ [50]

Kapitel 4

Design & Realisierung

In diesem Kapitel wird die Architektur der „RFID-Watch-Box“ und des „RFID-Configurators“ näher beleuchtet. Zuerst wird auf die konkrete Realisierung der Hardware eingegangen. Darauf folgend soll die gewählte Softwarearchitektur durch den Einsatz von Klassen- und Sequanzdiagrammen dem Leser näher gebracht werden.

4.1 Die Hardware

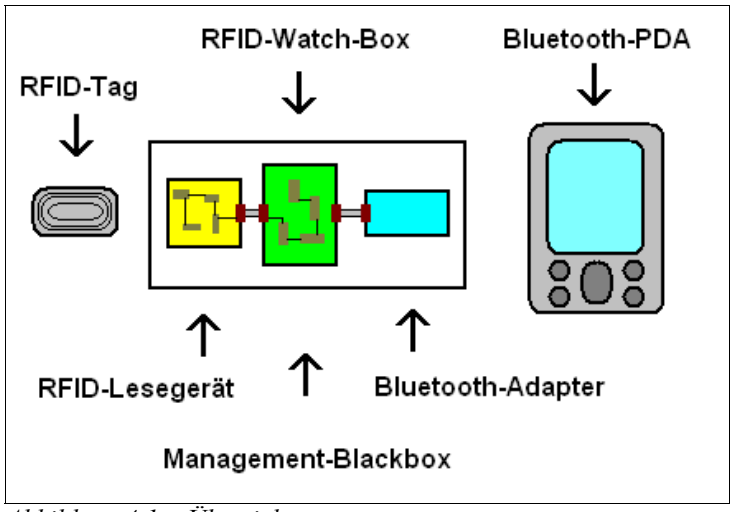

Abbildung 4.1a: Übersicht

Die Funktionalität des „RFID-Configurators“ soll durch einen handelsüblichen Bluetooth-PDA bereitgestellt werden. Die „RFID-Watch-Box“ besteht aus einem RFID-Lesegerät, einer „Management-Blackbox“ und einem Bluetooth-Dongle. Die „Management-Blackbox“ enthält die Logik, die für die Umgebungsüberwachung und das Einhalten der Regeln zuständig ist. Sie ist über jeweils eine RS232-Schnittstelle mit dem RFID-Lesegerät sowie mit dem Bluetooth-Adapter verbunden (siehe Abb. 4.1a).

4.1.1 RFID-Lesegerät

Als RFID-Lesegerät kommt der 13,56 MHz ISO-Reader der Firma Megaset zum Einsatz. Dieser RFID-Reader für Transponder nach ISO/IEC 15693 hat leider nur eine Reichweite von etwa 10 Zentimetern. Da im Computerlabor der Hochschule im Zeitraum der Bearbeitung der Diplomarbeit leider kein anderes Lesegerät zur Verfügung stand, muss diese Einschränkung für den Prototypen leider hingenommen werden. Im Kapitel 5 „Resümee und Ausblick“ wird noch einmal auf die Reichweitenproblematik eingegangen.

Abbildung 4.1.1a: ISO-Reader-Controller

Abbildung 4.1.1b: ISO-Reader-Box

4.1.2 „Management-Blackbox“

Die „Management-Blackbox“ ist quasi das Herz der „RFID-Watch-Box“. Sie besteht aus einem µController und einem seriellen EEPROM. Beides ist aus dem Hause Microchip Technology. Der µController und der EEPROM sind über einen I²C-Bus miteinander verbunden. Mit dem RFID-Lesegerät und dem Bluetooth-Adapter kommuniziert der µController über zwei RS232-Schnittstellen. Die RS232-Schnittstellen werden mit Maxim MAX233 Treibern betrieben. Schließlich ist noch eine In-Circuit-Programmierschnittstelle für den µController vorgesehen (Abb. 4.1.2a – Programmierschnittstelle nicht dargestellt).

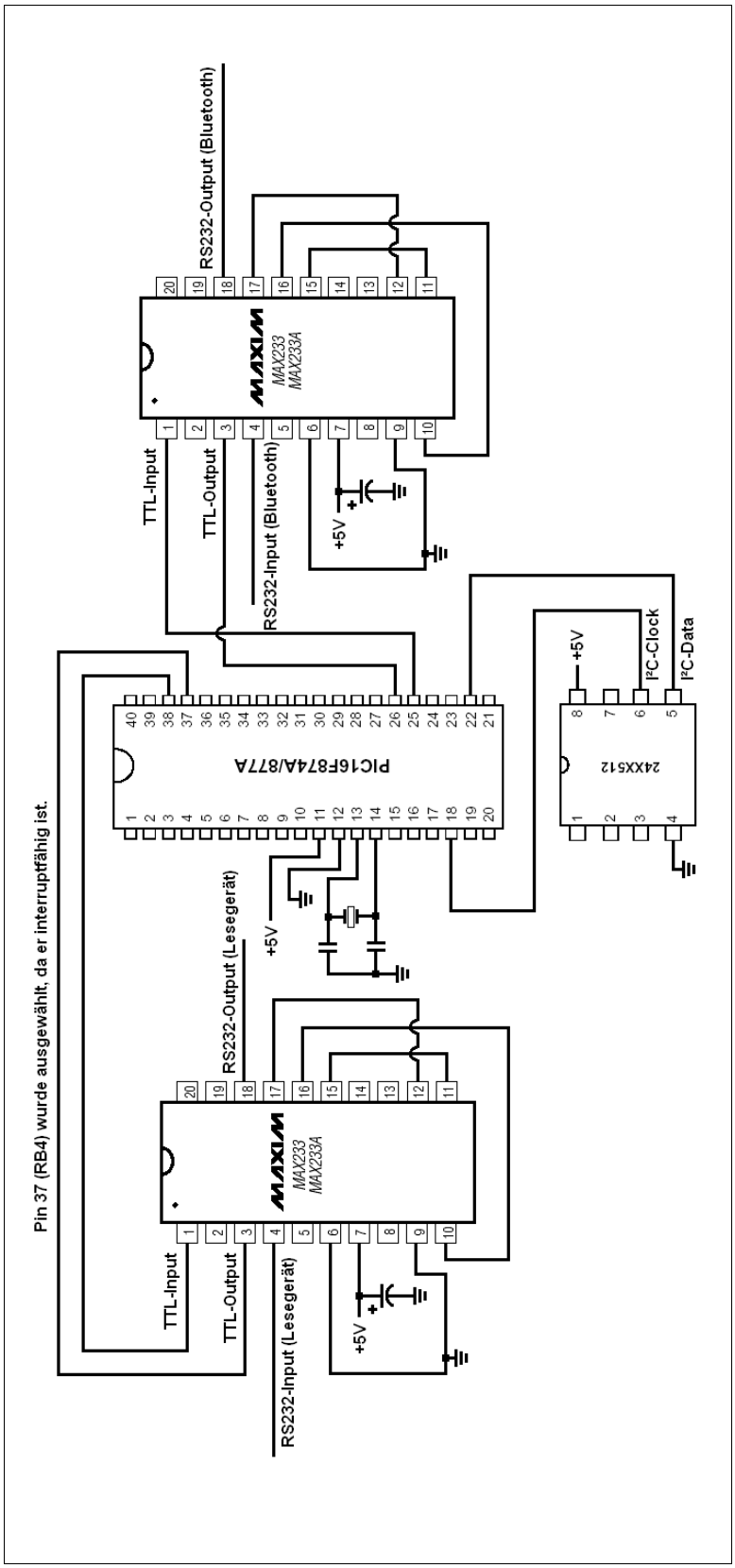

Abbildung 4.1.2a: Schaltung der Management-Blackbox

µController (Microchip 16F877A)

Für einen ersten Prototyp wurde der Microchip 16F877A µController ausgewählt. Den 16F877A zeichnet eine Vielzahl an I/O-Pins aus. Über den vorhandenen I²C-Anschluss[1] kann der externe serielle EEPROM direkt angesprochen werden. Sollten einem die internen Timer des µControllers nicht zur Zeitmessung genügen, könnte man auch noch eine RTC[2] an den I²C-Bus anschließen. Der µController besitzt leider nur einen einzigen USART[3] in Hardware, jedoch hat er genug Rechenleistung, um eine zweite RS232-Verbindung an anderen I/O-Pins des µControllers in Software zu betreiben. Man könnte zwar auch einen anderen µController auswählen, der stattdessen zwei USARTs in Hardware bietet, doch da der 16F877A bereits vorrätig war, wurde diese kleine Einschränkung erstmal akzeptiert. Die Vielzahl von nun noch freien I/O-Pins lässt auch noch viele Erweiterungsmöglichkeiten zu. Jedoch ist der IC durch seine große Anzahl an Pins leider auch in seinen Abmessungen relativ groß geraten, sodass man für eine miniaturisierte „Serienversion“ vielleicht lieber einen anderen µController auswählen sollte. Diese Überlegung wäre auch ratsam, zumal andere ebenfalls nutzbare µController auch preislich günstiger wären.

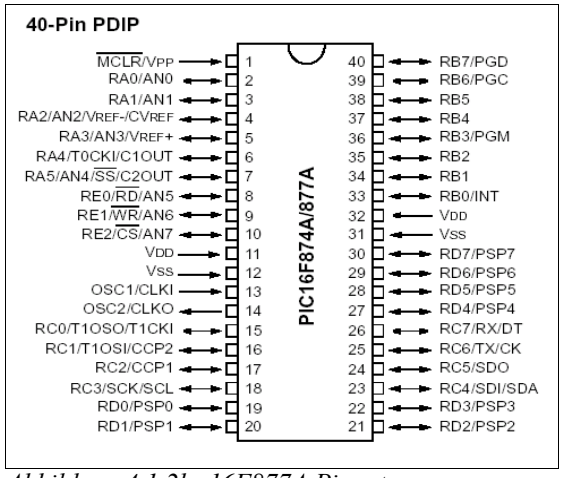

Abbildung 4.1.2b: 16F877A Pinout

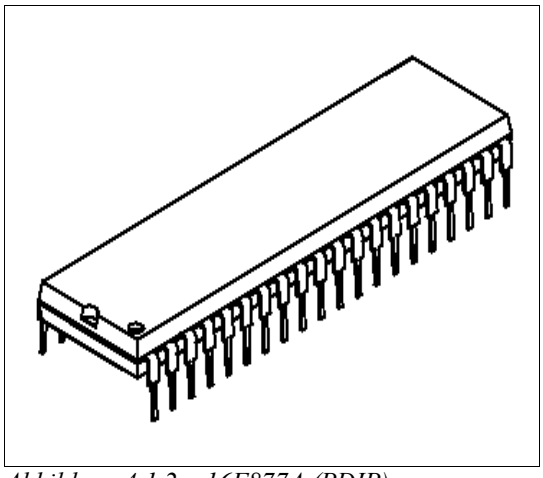
Abbildung 4.1.2c: 16F877A (PDIP)

1 I²C ist ein von Philips Semiconductor entwickelter und weit verbreiteter serieller Bus
2 Real Time Clock
3 Universal Asynchronous Receiver Transmitter

Serieller EEPROM (Microchip 24AA512)

Zur Speicherung der Zeitpunkte von Sichtungen der zu trackenden RFID-Tags und der auftretenden Regelverstöße, soll ein externer serieller EEPROM genutzt werden. Der ohnehin vorrätig gewesene Microchip 24AA512 mit seinen 512 kBit (64k x 8) Datenspeicher sollte für diese Aufgabe mehr als ausreichend sein. Er bietet mindestens 1.000.000 Lösch-/Schreibzyklen, so dass auch seine Lebenszeit ausreichend lang sein sollte. Wenn man annimmt, dass das Programm so geschrieben wird, dass es die Zeiten auf die Minute genau speichert, würde im Worst-Case-Szenario, welches aus einem dauerhaften Löschen und Neuschreiben der Uhrzeit im Minutentakt bestünde, der EEPROM auf alle Fälle länger als 340 Tage in seiner Funktion unbeeinträchtigt bleiben (1.000.000/2/60/24 Tage).

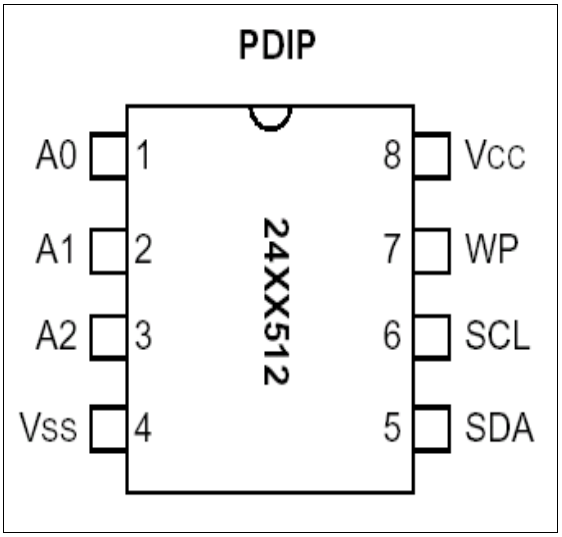

Abbildung 4.1.2d: 24AA512 Pinout

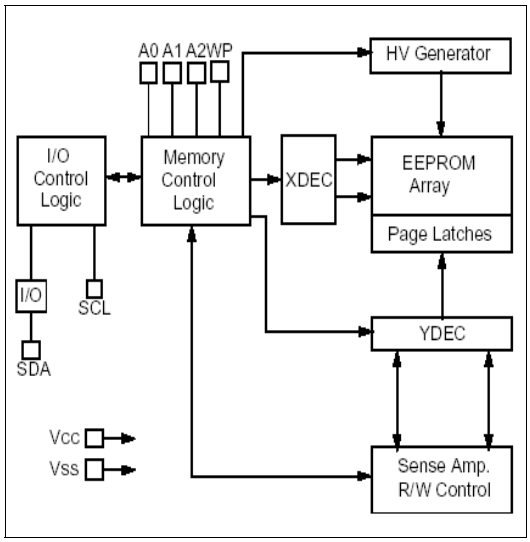

Abbildung 4.1.2e: 24AA512 Block Diagram

RS232-Schnittstellentreiber (Maxim MAX233)

Beim MAX233 handelt es sich um einen ganz normalen RS232-Schnittstellentreiber-Baustein, der im Gegensatz zum landläufig eingesetzten MAX232 ohne externe Kapazitäten auskommt. Der Vorteil liegt ganz klar in der Platzersparnis auf der Platine und in unserem Fall des Prototypen in etwas weniger Lötarbeit.

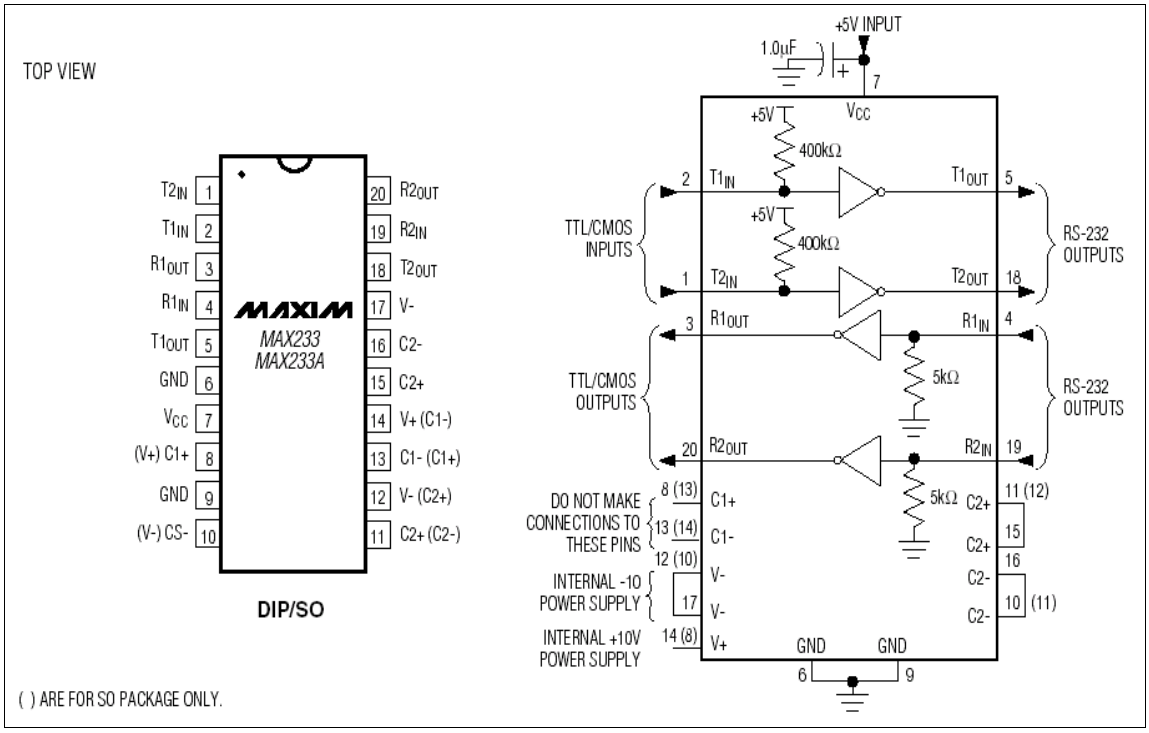

Abbildung 4.1.2f: MAX233 Pinout und typische Beschaltung

4.1.3 Bluetoothadapter

Die Datenverbindung zwischen „RFID-Watch-Box“ und „RFID-Configurator“ soll zeitgemäß und anwenderfreundlich auf kabellosem Wege erfolgen. Hier stellt sich eigentlich nur die Frage, ob es sich um eine Bluetooth- oder WLAN-Verbindung handeln soll. Andere Lösungen wie zum Beispiel eine FSK-modulierte Funkverbindung im 434 MHz ISM-Bereich scheiden von vornherein aus. Zwar würde eine solche Funkverbindung auf Seiten der „RFID-Watch-Box“ eine enorme Platz- und Energieersparnis bedeuten, da man statt dem hier verwendeten 16F877A µController und dem Bluetoothadapter einen rfPIC[1] wie den

1 Die rfPIC Serie beinhaltet bereits einen kompletten UHF-Sender, der ASK bzw. FSK modulierte Datenübertragung erlaubt. Als Empfänger kann zum Beispiel ein Microchip rfRXD0420, der nur einige wenige externe Komponenten benötigt, verwendet werden.

12F675F und einen einfachen Empfänger wie den rfRXD0420 verwenden könnte, doch da dieses auch eine hardwareseitige Änderungen auf Seiten des PDAs bedeuten würde, welcher ja als „RFID-Configurator“ dienen soll, ist dieser Weg inakzeptabel. Also bleibt nur noch die Wahl zwischen Bluetooth und WLAN. Beides ist bei dem im Labor der Hochschule verwendeten PDA vorhanden. Ein Blick auf den Energieverbrauch von aktuellen Bluetooth- und WLAN-Modulen fiel zugunsten von Bluetooth aus. Zudem hält die Entscheidung für Bluetooth die Möglichkeit offen, eine spätere Portierung des „RFID-Configurators“ auf ein Mobiltelefon mit Bluetoothschnittstelle zu realisieren. Und last but not least ist der Bluetooth-Standard von vornherein prädestiniert für eine wie hier vorliegenden Point-to-Point Peripherieanbindung, wohingegen WLAN mit seinen ausgeprägten Netzwerkeigenschaften eigentlich etwas anderes bietet, als die hier vorliegende Aufgabe erfordert. Stark vereinfacht ausgedrückt könnte man sagen, dass WLAN die kabellose Version von Ethernet ist, wohingegen Bluetooth eher als ein Ersatz für USB-Verbindung angesehen werden kann[1].

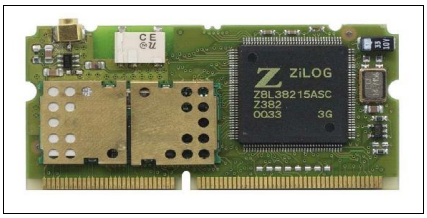

Abbildung 4.1.2g: Pico Core

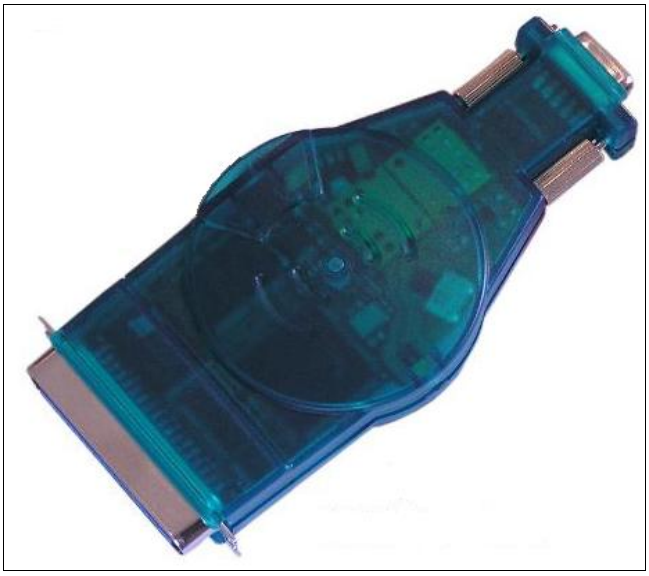
Abbildung 4.1.2h: Pico Plug

Konkret wird nun auf Seiten des Prototypen der „RFID-Watch-Box“ ein Pico Plug der Firma Sphinx-Electronics verwendet. Dieser relativ große Adapter könnte im Zuge einer gewünschten Miniaturisierung zum Beispiel durch einen Pico Core derselben Firma ersetzt werden. Der Pico Plug besitzt sowohl einen RS232- als auch einen Centronics-Anschluss. Nach einer einmaligen Konfiguration des Pico Plugs durch die dem Adapter beiliegende Software des Herstellers kann der Pico Plug im Zusammenhang mit einem zweiten Bluetooth-Adapter, der ebenfalls das auf RFCOMM aufsetzende Serial Port Profile unterstützt, eine normale serielle COM-Verbindung aufbauen.

1 Siehe dazu auch c't Heftausgabe 23/2003 Seite 142 „Firstclass Luftverkehr - Bluetooth vs. WLAN“ [42]

4.1.4 Bedarfs-GUI auf dem Windows Mobile 2003 PDA

Der „RFID-Configurator“ stellt dem Benutzer im Bedarfsfall ein GUI zur Interaktion mit dem RFID-Tracking-System zur Verfügung. In konkreten Fall kommt auf Seiten der Hardware ein HP iPAQ H5550 zum Einsatz. Dieser PDA wird standardmäßig mit Microsoft Windows Mobile 2003 und der Jeode JavaVM ausgeliefert. Seine technischen Daten sehen wie folgt aus:

Prozessor:	Intel Xscale mit 400 MHz
Speicher:	128 MB SDRAM und 48 MB Flash-ROM
Steckplatz:	SD-Steckplatz für SD, SDIO und MMC
Display:	3,8 Zoll TFT, 240x320 Pixel mit 65k Farben
I/O:	Seriell, USB-Client, IrDA, WLAN und Bluetooth

Abbildung 4.1.4a: PDA mit RFID-Configurator Software

4.2 Die Software

Im folgenden Kapitel wird auf das Design und die Realisierung der Software eingegangen. Aufgrund der Systemaufteilung in „RFID-Configurator“ (PDA) und „RFID-Watch-Box“ werden diese beiden Teile auch hier in zwei getrennten Kapiteln behandelt:

- **Kapitel 4.2.1** der „RFID-Configurator“ auf dem PDA
- **Kapitel 4.2.2** die „Management-Blackbox“ in der „RFID-Watch-Box“

Auf Seiten der Software für den „RFID-Configurator“ wurde ein objektorientiertes Design in Java gewählt. Hierin wird jede Regel und jeder RFID-Tag durch ein eigenes Objekt repräsentiert. Die Verwaltung dieser Regel- und Tag-Objekte übernimmt eine Instanz der Klasse *RFIDRuleManager* beziehungsweise *RFIDTagManager*. Zur Kommunikation mit der „RFID-Watch-Box“ ist ein Stellvertreter-Objekt vorgesehen, welches eine Instanz der Klasse *RFIDWatchBoxProxy* ist (siehe Abb. 4.2.1.2b). Die tatsächliche „RFID-Watch-Box“ wurde in der Programmiersprache C gelöst. Hierbei basiert die Software auf der Implementation eines Automaten (siehe Abb. 4.2.2.2a).

4.2.1 Der „RFID-Configurator“ auf dem PDA

4.2.1.1 Die Vorzüge von PersonalJava gegenüber PocketPC.NET

Zu entscheiden war, ob die Software auf Basis des etwas betagten PersonalJava oder doch lieber dem aktuelleren PocketPC.NET entwickelt werden sollte. Eine JavaVM für PersonalJava ist mit der durch HP mit dem PDA mitgelieferten JeodeVM vorhanden, auch ist eine PocketPC.NET-VM inklusive kompletter Entwicklungsumgebung kostenlos von Microsoft beziehbar. Der Wunsch die Software später jedoch nicht nur auf Windowssystemen einsetzen zu können, sondern auch auf PDAs auf Linuxbasis, wie sie seit neusten im Labor der Hochschule ebenfalls vorhanden sind, ließ die Entscheidung für Java nahe liegen. Die Möglichkeit, den „RFID-Configurator“ durch die Wahl von Java, später auch leicht auf ein Java MIDP 2.0 Mobiltelefon portieren zu können, gab letztendlich den endgültigen Ausschlag für die Entscheidung Java zu verwenden.

4.2.1.2 Klassendiagramm

Die Software des „RFID-Configurators“ ist grundsätzlich in zwei Bereiche aufgeteilt: Zum einen ist da die eigentliche Management-Logik und zum anderen das für die Darstellung zuständige GUI. Der Zugriff des GUIs auf die eigentliche Management-Logik erfolgt ausschließlich über eine genau definierte Schnittstelle, die durch die Klasse *RFIDManager* gestellt wird (siehe Abb. 4.2.1.2a). Sollte man nun für das Programm eine neue Oberfläche für zum Beispiel ein MIDP 2.0 Mobiltelefon gestalten wollen, muss ausschließlich der darstellende Teil, der auf eben diese Schnittstelle zugreift, ausgewechselt werden. Auf Abbildung 4.2.1.2b ist das gesamt Klassendiagramm des „RFID-Configurators“ zu sehen (verkleinerte Fassung, Vollbild im Anhang).

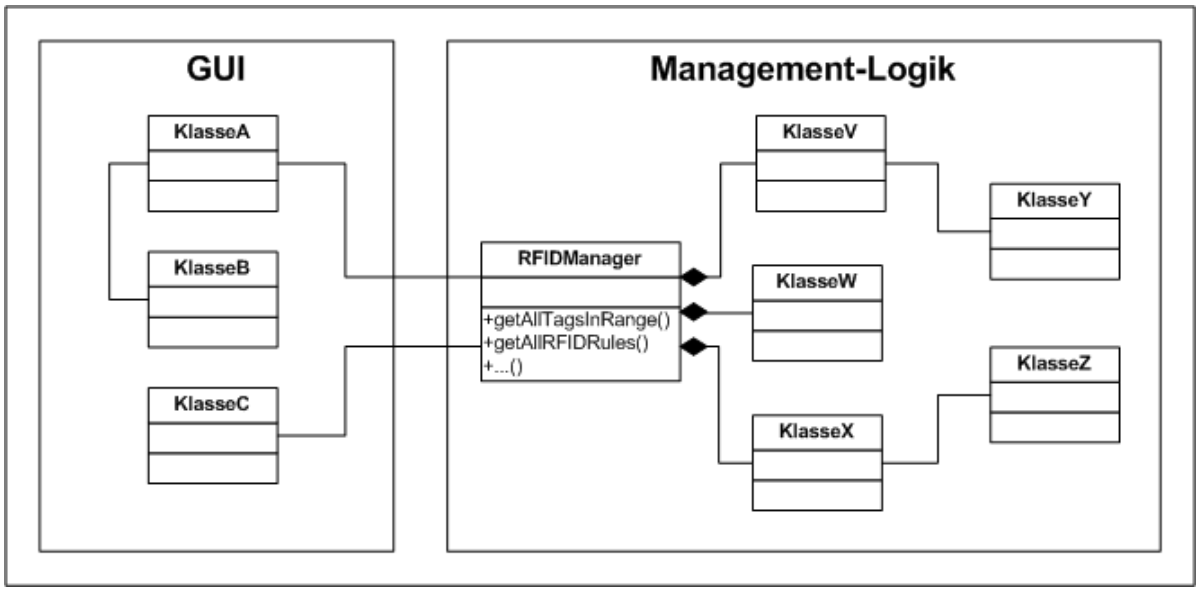

Abbildung 4.2.1.2a: Freie Stilisierung der Zweiteilung von GUI und Management-Logik

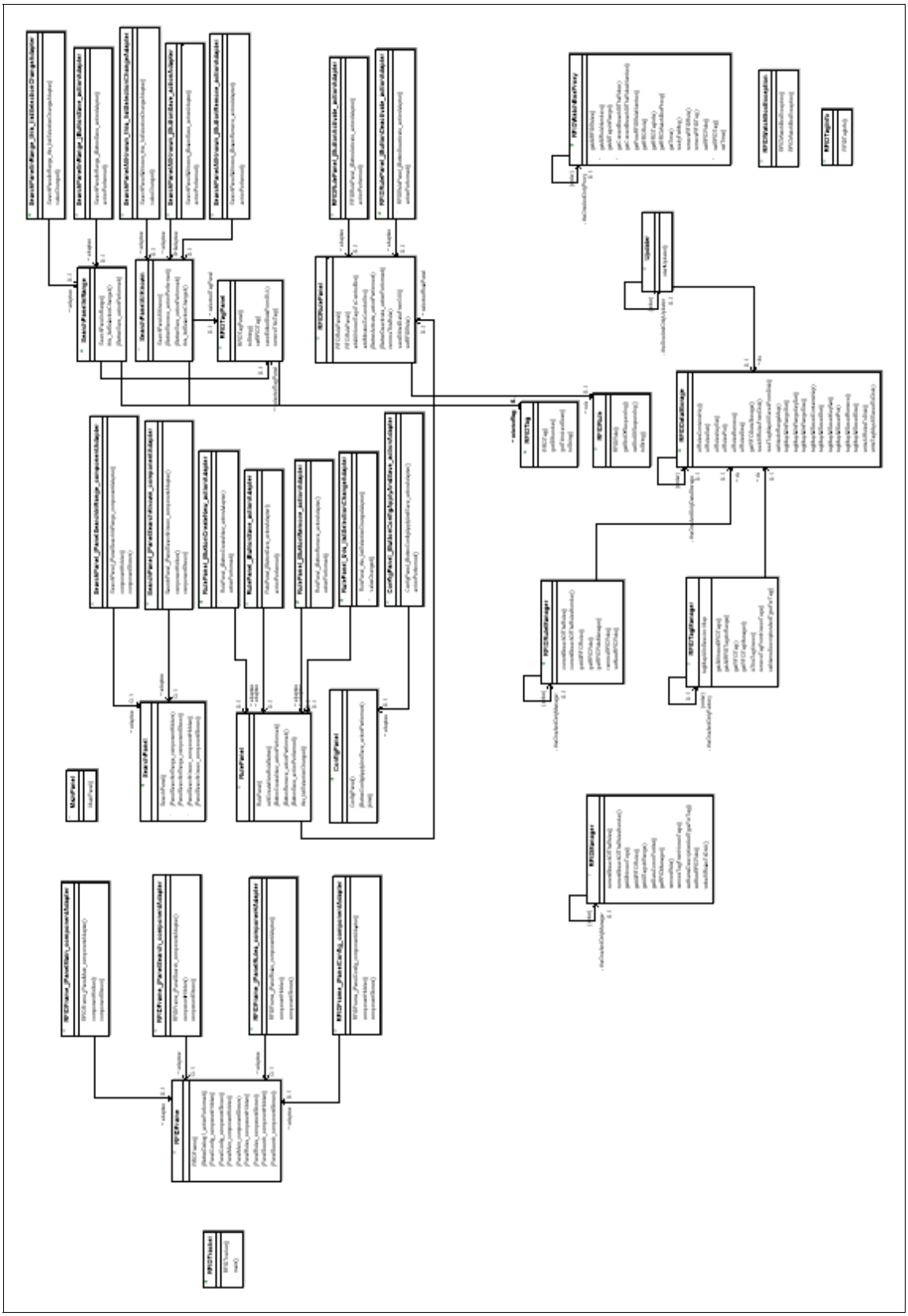

Abbildung 4.2.1.2b: Klassendiagramm (miniaturisierte Ansicht – Vollbild siehe Anhang)

4.2.1.3 Sequenzdiagramme

Um die Funktionsweise des Programms zu verdeutlichen, werden folgend einzelne Abläufe in der Software als Sequenzdiagramme dargestellt und textuell begleitend beschrieben. Aufgrund des Umfangs der Software kann leider nur eine Auswahl getroffen werden, so dass nicht auf alle vorhandenen Funktionen eingegangen wird.

Diese Abläufe werden nun folgend beschrieben:

a. Abfrage der aktuellen Regelverstöße

b. Akzeptieren eines aktuellen Regelverstoßes

c. Abfrage von „Tags In Range“

d. Abfrage der genauen Daten eines einzelnen RFID-Tags

e. speichern eines RFID-Tags

f. der Datenabgleich zwischen „RFID-Configurator“ und „RFID-Watch-Box“

g. das dynamische Instantiieren einzelner Programmteile

a. Abfrage der aktuellen Regelverstöße

Nach dem Start des Programms wird dem Benutzer als erstes die Main-Registerkarte angezeigt. Hier wird er über aktuell anstehende Regelverstöße informiert und bekommt die Möglichkeit, diese im Nachhinein zu akzeptieren. Dieser Programmteil ist das erste, was der Benutzer angezeigt bekommt, weil der Programmstart nach einer akustischen Warnung über einen Regelverstöß durch die „RFID-Watch-Box“ wohl die häufigste Anwendung für den „RFID-Configurator“ sein wird. Da in diesem Fall der akustische Alarm erst aufhört, nachdem der Benutzer über den „RFID-Configurator“ diesen Regelverstoß im Nachhinein akzeptiert hat, dürfte der Benutzer auch sehr dankbar darüber sein, wenn er sich nicht erst lange durch viele Untermenüs hangeln muss, um den Alarm abstellen zu können.

Hier kommt nun das Sequenzdiagramm, wie das GUI die eigentliche Management-Logik über den Stand der aktuellen Regelverstöße befragt. Zur Verdeutlichung sei hier angemerkt, dass die Klasse *RFIDRuleManager* nach dem Singleton-Pattern implementiert ist. Da wir annehmen, dass diese Abfrage direkt beim Programmstart geschieht, hat noch kein anderes Objekt ein *RFIDRuleManager* Objekt angefordert, sodass dieses erst noch instantiiert werden muss. Bei allen weiteren Anforderungen ist die Instanz bereits vorhanden.

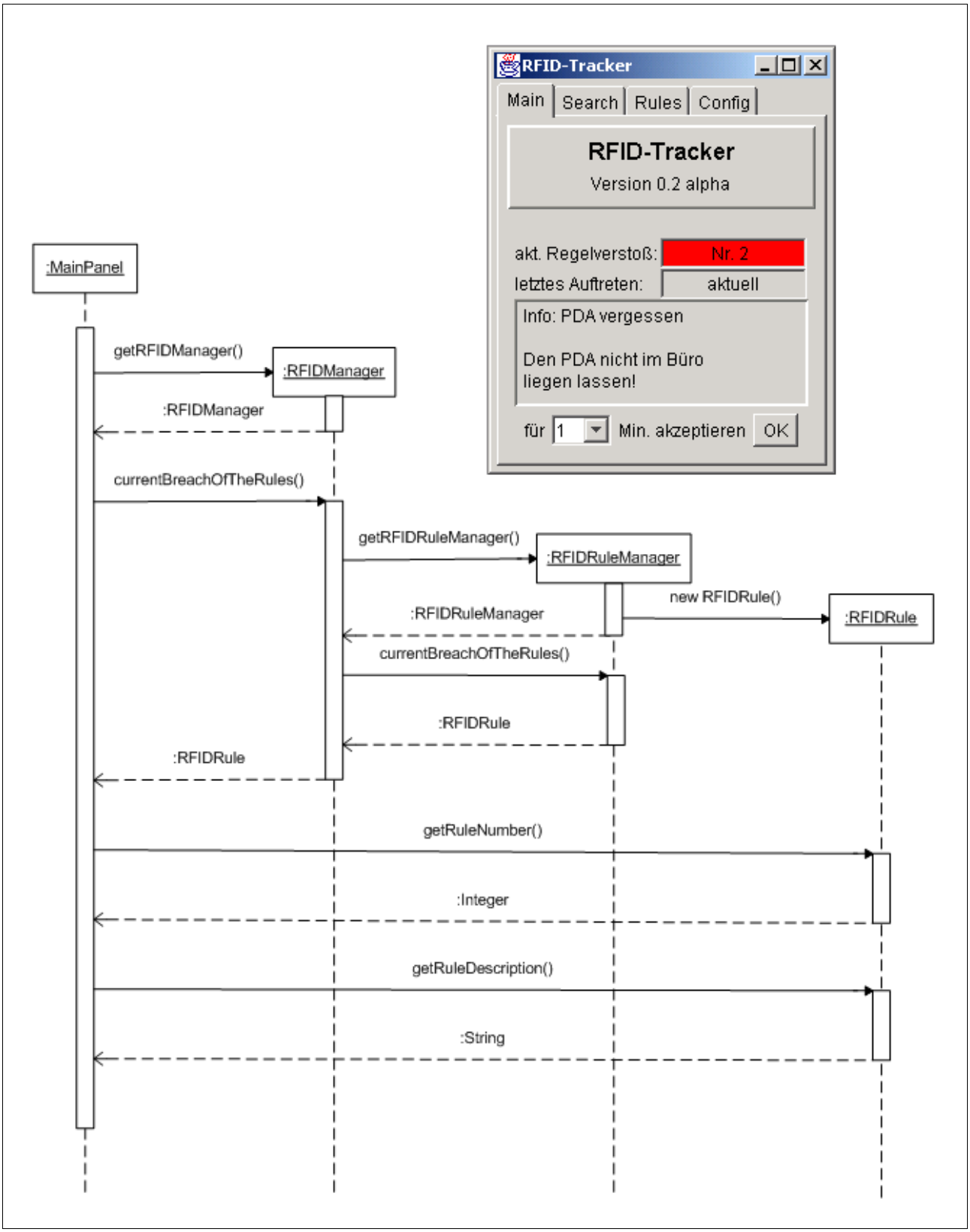

Abbildung 4.2.1.3a: Sequenzdiagramm „Abfrage der aktuellen Regelverstöße“

b. Akzeptieren eines aktuellen Regelverstoßes

Nachdem nun, wie eben beschrieben, die aktuellen Regelverstöße abgefragt wurden, wird nun einer von ihnen angezeigt, natürlich nur falls mindestens einer vorhanden ist. Der Benutzer muss nun entscheiden, für wie lange er die Regel suspendieren möchte, um dadurch den Regelverstoß im Nachhinein zu akzeptieren. Diese Suspendierungsaufforderung schickt das GUI dann an das *RFIDManager* Objekt und löst folgende Sequenz aus. Es wird die aktuelle Zeit plus dem Suspandierungszeitraum aufgerundet auf die nächste volle Minute als Endzeitpunkt für die Suspendierung gesendet:

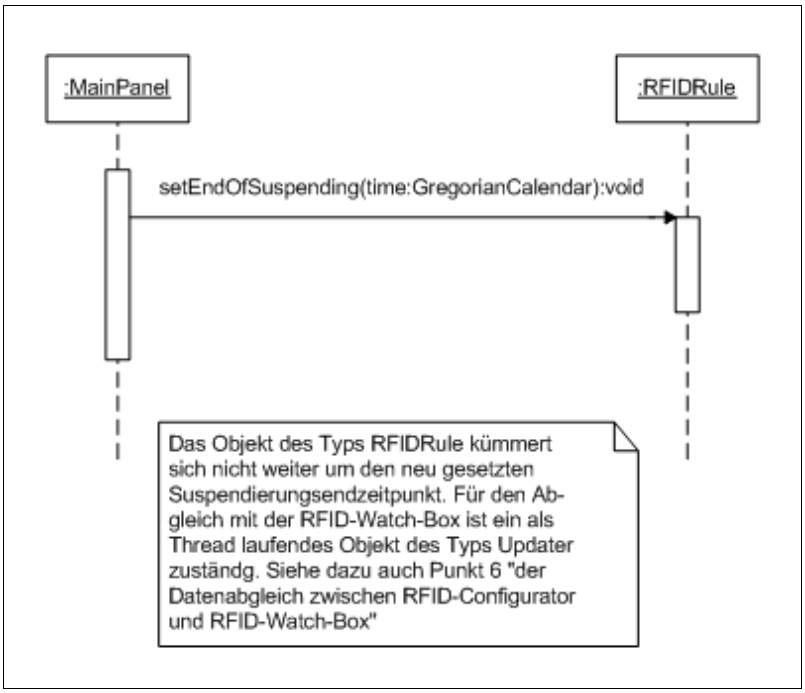

Abbildung 4.2.1.3b: Sequenzdiagramm „Akzeptieren eines aktuellen Regelverstoßes“

c. Abfrage von „Tags In Range“

Unter der Search-Registerkarte findet der Anwender eine weitere Registerkarte mit dem Namen „Tags In Range“. Hier fordert das GUI von dem *RFIDManager* Objekt Informationen über in Reichweite befindlicher RFID-Tags an. Dieses läuft wie folgt ab:

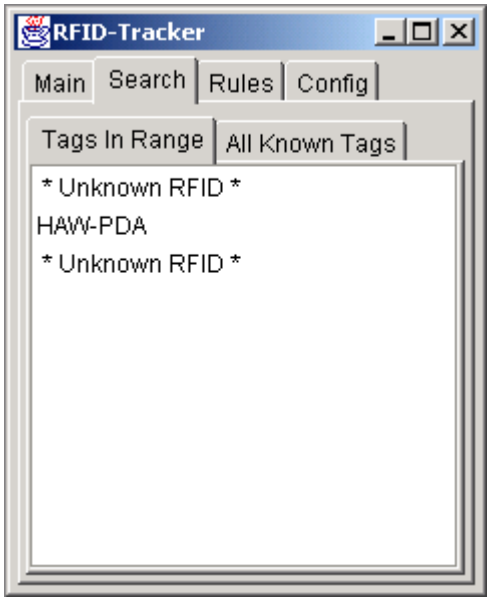

Abbildung 4.2.1.3c: „Tags In Range“

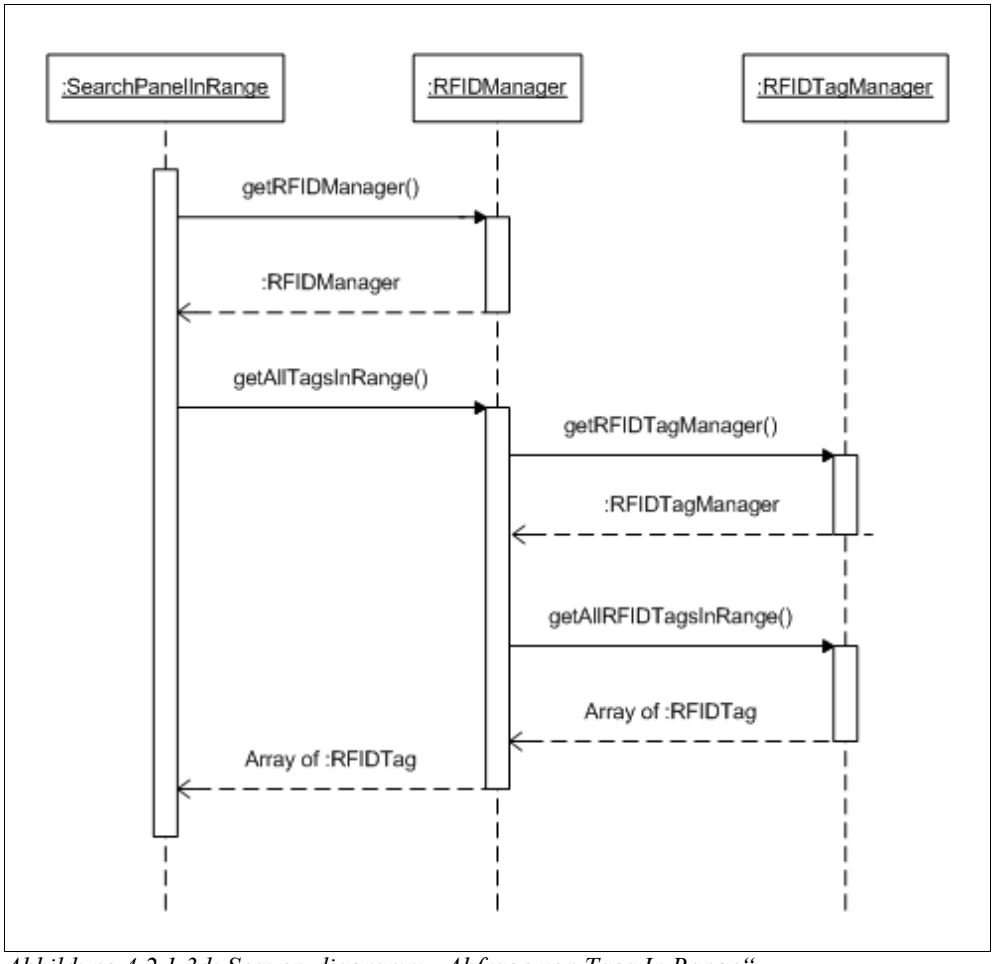

Abbildung 4.2.1.3d: Sequenzdiagramm „Abfrage von Tags In Range“

d. Abfrage der genauen Daten eines einzelnen RFID-Tags

Da das GUI nun, wie unter „c“ beschrieben, die gefundenen RFID-Tags anzeigt, kann der Benutzer auf eines dieser Tags tippen und es wird innerhalb des GUI ein weiteres Objekt eingeblendet, welches von einer erweiterten Klasse von *JPanel* stammt. Diese erweiterte Klasse nennt sich *RFIDTagPanel.* Wie mit dem Objekt vom Typ *RFIDTagPanel* das RFID-Tag nun angezeigt wird, stellt folgende Sequenz dar (einige Methodenaufrufe wurden hier der Übersichtlichkeit wegen weggelassen):

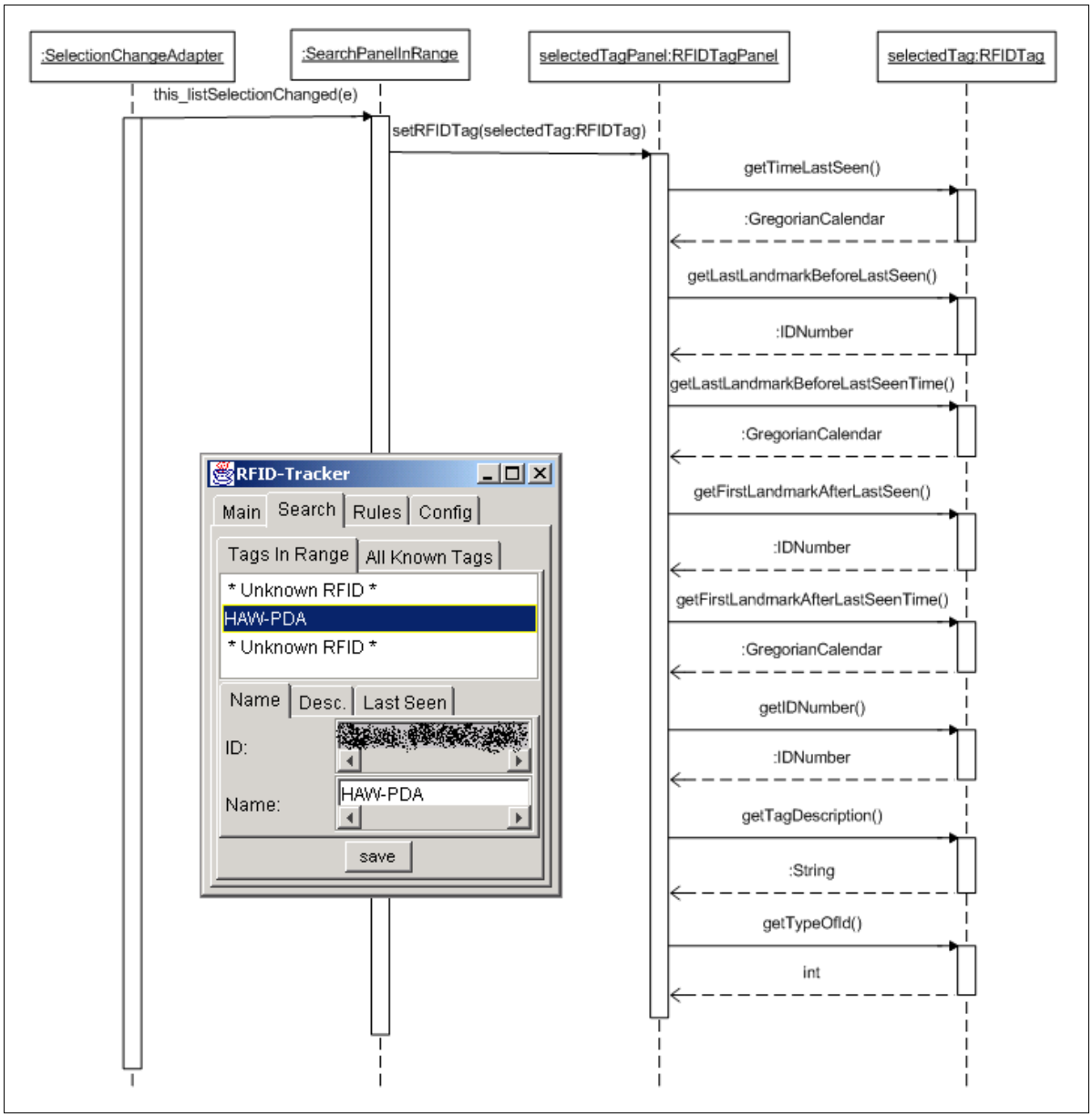

Abbildung 4.2.1.3e: Sequenzdiagramm „Abfrage der genauen Daten eines einzelnen RFID-Tags“

e. Speichern eines RFID-Tags

Durch den unter „d.“ beschriebenen Vorgang wird nun der Informationsgehalt eines *RFIDTag* Objektes über ein *RFIDTagPanel* Objekt für den Benutzer zugänglich gemacht. Einzelne Daten wie zum Beispiel der Name des RFID-Tags können darin editiert werden. Nach dem Editieren müssen die gemachten Änderungen noch abgespeichert werden. Dazu sendet das GUI mit Zuhilfenahme des zu ändernden *RFIDTag* Objektes diese Daten wie folgt an die Management-Logik:

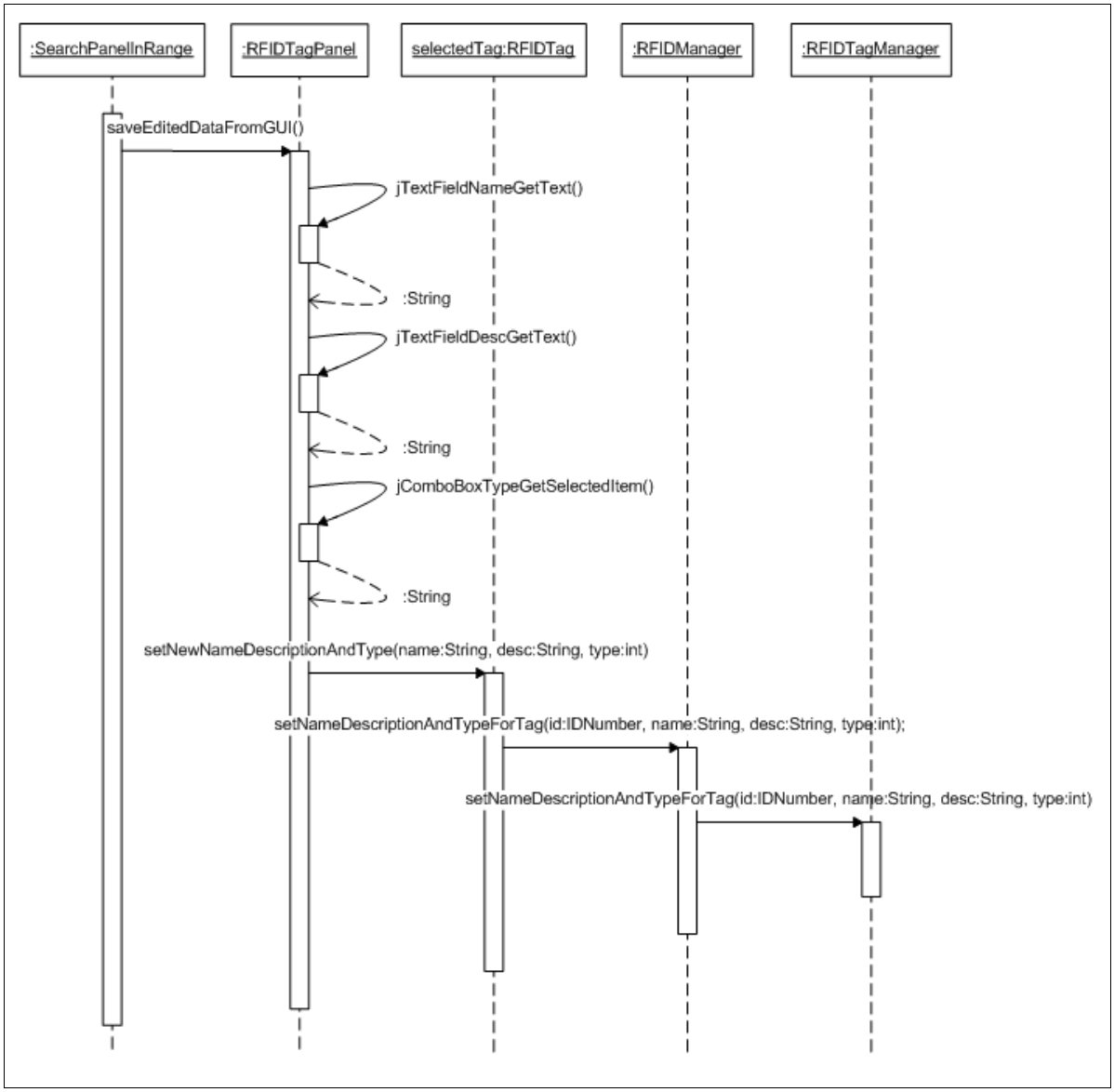

Abbildung 4.2.1.3f: Sequenzdiagramm „speichern eines RFID-Tags“

f. Der Datenabgleich zwischen „RFID-Configurator“ und „RFID-Watch-Box“

Der aufmerksame Leser mag nun bemerkt haben, dass in den vorhergehenden Sequenzdiagrammen nie ein direkter Kontakt mit der „RFID-Watch-Box“ stattgefunden hat. Das ist richtig. Innerhalb des „RFID-Configurators“ gibt es drei wichtige Maps. Die erste Map beinhaltet alle dem System bekannten und somit auch zu trackenden RFID-Tags, die zweite Map verwahrt die RFID-Tags, die aktuell in Reichweite des Lesegeräts sind und die dritte Map führt eine Liste aller aufgesetzten Regeln. Zudem ist noch die Anzahl der aktuellen Regelverstöße gespeichert. Wenn diese Anzahl größer Null ist, ist auch eine dieser Regeln, gegen die verstoßen wurde, hinterlegt. Alle diese Informationen werden in einer Instanz der Klasse *RFIDDataStorage* verwaltet. Alle Methodenaufrufe auf dieses *RFIDDataStorage*-Objekt sind synchronized. Ein Objekt vom Typ *Updater* kümmert sich als parallel laufender Thread darum, dass der Datenbestand zwischen „RFID-Configurator“ und „RFID-Watch-Box“ konsistent bleibt beziehungsweise wieder wird, wenn Diskrepanzen auftreten. Dafür sorgt er indem er folgendes macht:

1. In der Map für zu trackende RFIDs:
   - falls ein RFID-Tag auf dem „RFID-Configurator“ hinzukommt, muss er ebenfalls auf der „RFID-Watch-Box“ hinzukommen
   - falls ein RFID-Tag auf dem „RFID-Configurator“ entfernt wurde, muss er ebenfalls auf der „RFID-Watch-Box“ entfernt werden
   - falls eine RFID-Tag-Information, welche durch die „RFID-Watch-Box“ gesammelt wurde, anders bzw. veraltet im „RFID-Configurator“ zu finden ist, muss dieser Informationseintrag im „RFID-Configurator“ erneuert werden.

2. In der Map für aktuell in Reichweite befindliche RFID-Tags
   - die Liste der Tags in Reichweite aus der „RFID-Watch-Box“ ist maßgebend, sollte die Aufstellung im „RFID-Configurator“ abweichen, wird dort die Liste aus der „RFID-Watch-Box“ übernommen

3. In der Map für alle Regeln:
   - falls eine Regel auf dem „RFID-Configurator“ hinzukommt, muss sie ebenfalls auf der „RFID-Watch-Box“ hinzukommen
   - falls eine Regel auf dem „RFID-Configurator“ entfernt wurde, muss sie ebenfalls auf der „RFID-Watch-Box“ entfernt werden
   - falls eine Regel auf dem „RFID-Configurator“ geändert wurde, muss sie ebenfalls auf der „RFID-Watch-Box“ geändert werden

4. Daten zu aktuellen Regelverstößen:
   - Anzahl aktueller Regelverstöße in der RFID-Watch-Box ist maßgebend und wird im RFID-Configurator übernommen. Wenn die Anzahl größer Null ist, wird eine dieser Regeln, gegen die verstoßen wurde, aus der RFID-Watch-Box abgefragt und im RFID-Configurator übernommen

Dieser Weg über einen Thread, der die Datenbestände fortwährend abgleicht, mag nicht der Eleganteste sein, schon da dadurch mit einhergeht, dass Veränderungen auf der einen Seite des Systems erst mit etwas Verzögerung auf der anderen Seite ankommen. Jedoch erleichtert es den Programmieraufwand für die Fehlerbehandlung von Kommunikationsproblemen zwischen „RFID-Configurator“ und „RFID-Watch-Box“ erheblich. In Hinblick auf die maximale Bearbeitungszeit für die Diplomarbeit wurde diese suboptimale Lösung als akzeptabel hingenommen. Damit der Benutzer weiß, wann der Konfigurationsstand auf dem „RFID-Configurator“ mit dem auf der „RFID-Watch-Box“ übereinstimmt, ertönt im „RFID-Configurator“ nach jedem Update ein kurzer, leiser Ton. Wenn der Benutzer zwischen zwei Tönen keine Veränderung vornimmt, weiß er, dass beide Konfigurationsstände übereinstimmen.

Hier nun zur Veranschaulichung der Vorgang wie das *Updater* Objekt die Map für die aktuell in Reichweite befindlichen RFID-Tags mit der „RFID-Watch-Box“ beziehungsweise dessen Proxy-Objekt abgleicht:

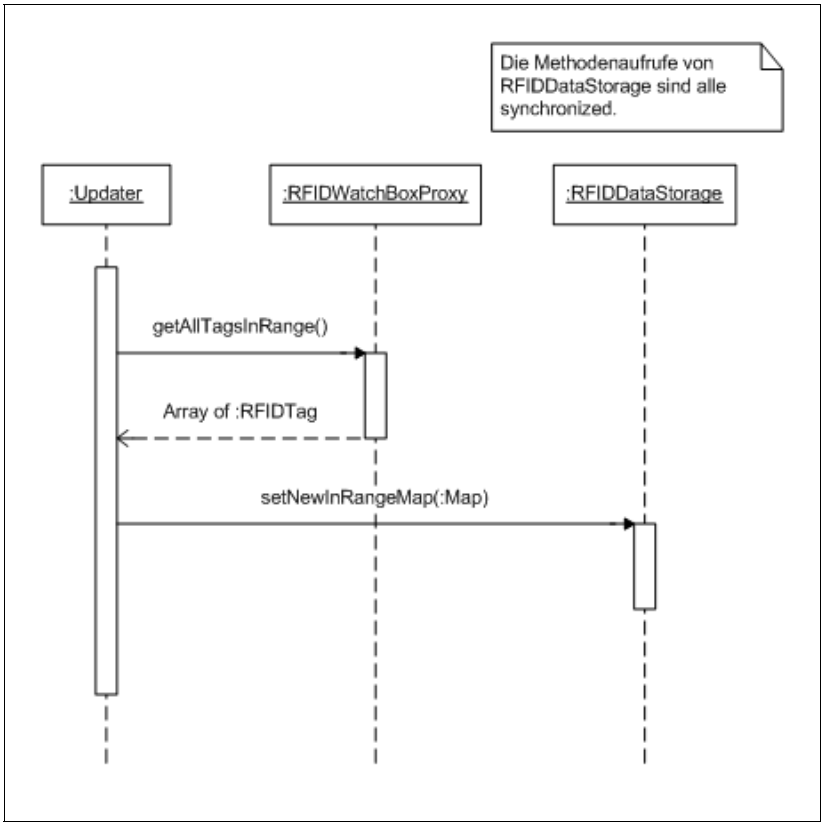

Abbildung 4.2.1.3g: Sequenzdiagramm „Updaten der Tags in Reichweite“

g. Das dynamische Instantiieren einzelner Programmteile

Im Zuge des Testens der Software wurde festgestellt, dass der Programmstart auf einem üblichen PC zwar zügig vonstatten ging, jedoch verhielt es sich auf dem iPAQ PDA ganz anders. Die Zeit für einen Programmstart belief sich dort auf fast eine Minute. Um dieses Problem zu lösen, wurde das Programm derart verändert, dass beim Programmstart nicht mehr alle Programmobjekte sofort instantiiert werden, sondern erst wenn diese benötigt werden. Zum Beispiel sind die einzelnen Registerkarten in der *jTabbedPane* eigentlich leere *JPanel*s (siehe Abbildung 4.2.1.3h). Wenn jedoch eins dieser *JPanel*s sichtbar wird, werden die benötigten Objekte instantiiert und auf ihre Position gebracht (siehe Abbildung 4.2.1.3i). Die Programmstartzeit konnte so auf 15 Sekunden reduziert werden. Auf aktuelleren PDA-Systemen mit mehr Rechenleistung sollte es natürlich noch etwas schneller gehen.

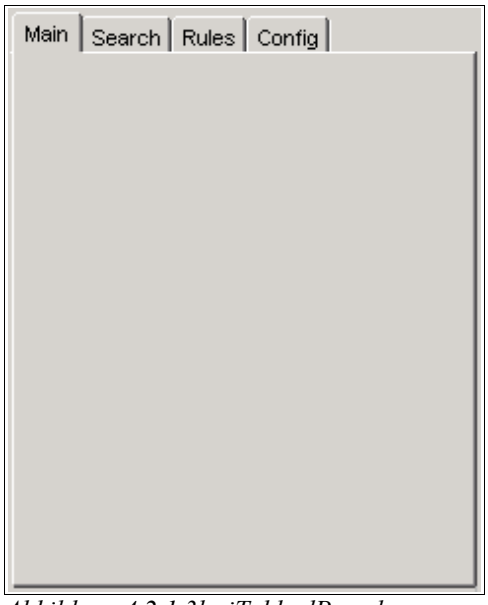

Abbildung 4.2.1.3h: jTabbedPane leer

Abbildung 4.2.1.3i: jTabbedPane geladen

Hier nun das Sequenzdiagramm für den Fall, dass die Registerkarte „Rules“ sichtbar wird:

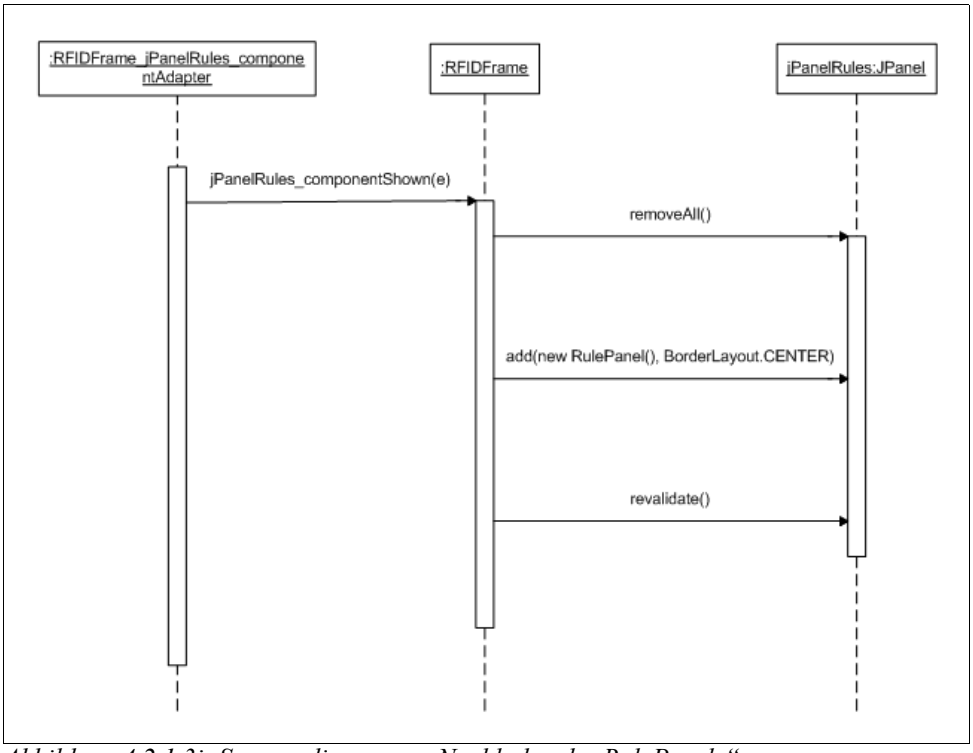

Abbildung 4.2.1.3j: Sequenzdiagramm „Nachladen des RulePanels“

4.2.1.4 Anmerkungen zu verwendeten Bibliotheken

Swing 1.1.1

Für die Gestaltung der Programmoberfläche wurde Swing gewählt. Ein Problem stellte dar, dass das verwendete PersonalJava leider nur JDK 1.1.8 voll unterstützt und hier noch kein Swing Einzug gehalten hatte. Die JFC 1.1 beinhalten jedoch ein Swing in Version 1.1.1, welches eigentlich zu PersonalJava kompatibel sein sollte. Leider gibt es jedoch einen „Bug“, der die Zusammenarbeit mit PersonalJava verhindert. Zum Glück gibt es aber auch eine Anleitung, wie man die Swing 1.1.1 Bibliothek so patchen kann, dass sie auch mit PersonalJava läuft [43] [44].

CEJavaComm 0.8pre1

Um aus Java heraus auf serielle RS232 und parallele IEEE1284 Schnittstellen zugreifen zu können, bietet Sun für die Betriebssysteme Solaris und Windows die Java Communications API an. Wer unter anderen Betriebssystemen auf eben diese Schnittstellen zugreifen möchte, muss sich nach Bibliotheken von Dritten umsehen. Im Internet findet man viele Implementationen der Java Communications API für diverse Betriebssysteme. Für Windows CE bzw. Windows Mobile bieten sich zum Beispiel CEJavaComm [45] und TXRXCommDriver [46] [47] als Lösung an. In diesem Dimplomarbeitsprojekt wurde die CEJavaComm Bibliothek in der Version 0.8pre1 verwendet. Gegenüber der Version 0.7 hat die 0.8pre1 den Vorteil, dass die immer wiederkehrende Fehlermeldung beim Aufzählen der vorhandenen Anschlüsse statt in einer MessageBox auf System.out angezeigt wird, wo es nicht weiter stört. Der Autor schreibt zwar auf seiner Homepage, dass sich die Bibliothek in einem frühen Stadium befindet, jedoch sind im Rahmen dieser Diplomarbeit keinerlei Fehler aufgetreten, die nachweislich auf diese Bibliothek zurückzuführen waren. Der Autor von CEJavaComm gibt seine Bibliothek leider nur für nicht kommerzielle Nutzung frei, sodass für solche Anwendungen schon deshalb lieber auf die TXRXCommDriver Bibliothek ausgewichen werden sollte. TXRXCommDriver steht unter der GNU Lesser General Public License.

4.2.2 Die „Management-Blackbox“ in der „RFID-Watch-Box“

Das RFID-Lesegerät der Firma Megaset wird bereits mit einer Firmware ausgeliefert, die es durch eine klar strukturierte Kommandosprache erlaubt, über eine RS232-Schnittstelle mit in Reichweite befindlichen RFID-Tags zu kommunizieren. Die eigentliche Überwachung und Protokollierung von RFID-Tag Sichtungen muss hingegen in einer selbst programmierten Systemkomponente stattfinden. Die Software dazu soll auf dem µController ausgeführt werden, welcher in der Beschreibung der „Management-Blackbox“ in Kapitel 4.1.2 zu finden ist. Die im Rahmen dieser Diplomarbeit gemachte erste Realisierung der Software für die „Management-Blackbox“ erfüllt nur einen Teil der Anforderungen, die im Kapitel 3.2 „Analyse der RFID-Watch-Box“ festgelegt wurden. Obwohl also der Prototyp nicht den gesamten Funktionsumfang enthält, wird hier im schriftlichen Teil die gesamte notwendige Design- und Realisierungsarbeit beschrieben. Zudem wurde in der tatsächlichen Realisierung die ständige Durchsuchung der Umgebung nach RFID-Tags so bewerkstelligt, dass versucht wird adressierte Kommandos an alle bekannten IDs zu senden. Wenn auf ein solches adressiertes Kommando eine Antwort erfolgt, gilt dieses RFID-Tag als in Reichweite befindlich. Ein unbekanntes RFID-Tag wird nur erkannt, wenn es als einziges Tag in Lesereichweite ist. Die Aufgabe der Umgebungsüberwachung musste leider derart gelöst werden, da das verwendete RFID-Lesegerät in seiner Firmwareversion 1.4 noch keine Anticollision unterstützt [48].

4.2.2.1 Der CCS-C Compiler

Da bereits während des Praxissemesters gute Erfahrungen mit dem CCS-C PCW Compiler gemacht wurden, ist er auch hier zum Einsatz gekommen. Leider ist die Studentenversion des Compilers auf den 16F876/16F877A beschränkt und die normale Vollversion ist mit 350 US-Dollar doch recht kostspielig. Bei der Verwendung von anderen µControllern sollte darüber nachgedacht werden, vielleicht doch lieber auf einen anderen Compiler auszuweichen. Hier reichte jedoch die Studentenversion vorerst aus.

Der Compiler bringt bereits seine eigenen IDE mit (siehe Abbildung 4.2.2.1a). Mit der CCS-C IDE ist man in der Lage mit dem so genannten „New Projekt Wizard“ schnell und einfach ein neues Projekt anzulegen (siehe Abbildung 4.2.2.1b). Sehr nützlich ist auch die Statistik-Funktion, die einen immer im Auge behalten lässt, ob noch genug Programm-ROM und vor allem RAM vorhanden ist. So ist man sehr gut im Stande, Entscheidungen über ein besseres Haushalten mit den vorhandenen Ressourcen zu treffen (siehe Abbildung 4.2.2.1c).

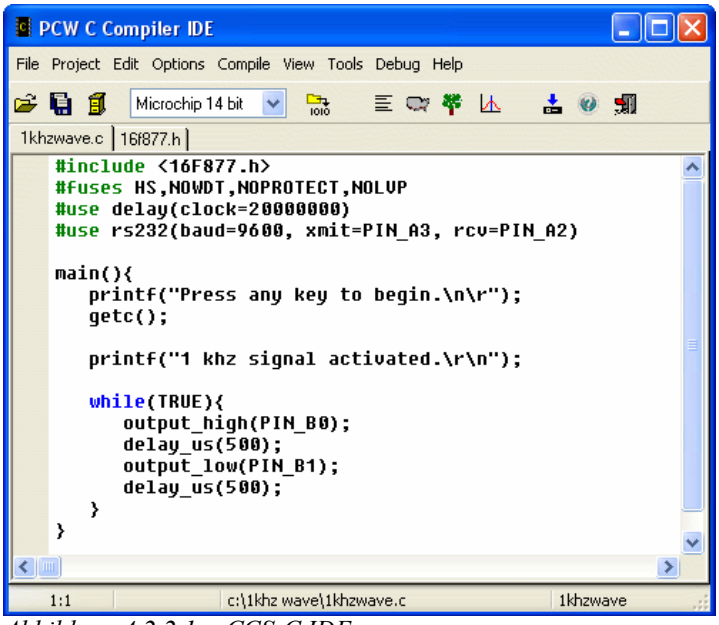

Abbildung 4.2.2.1a: CCS-C IDE

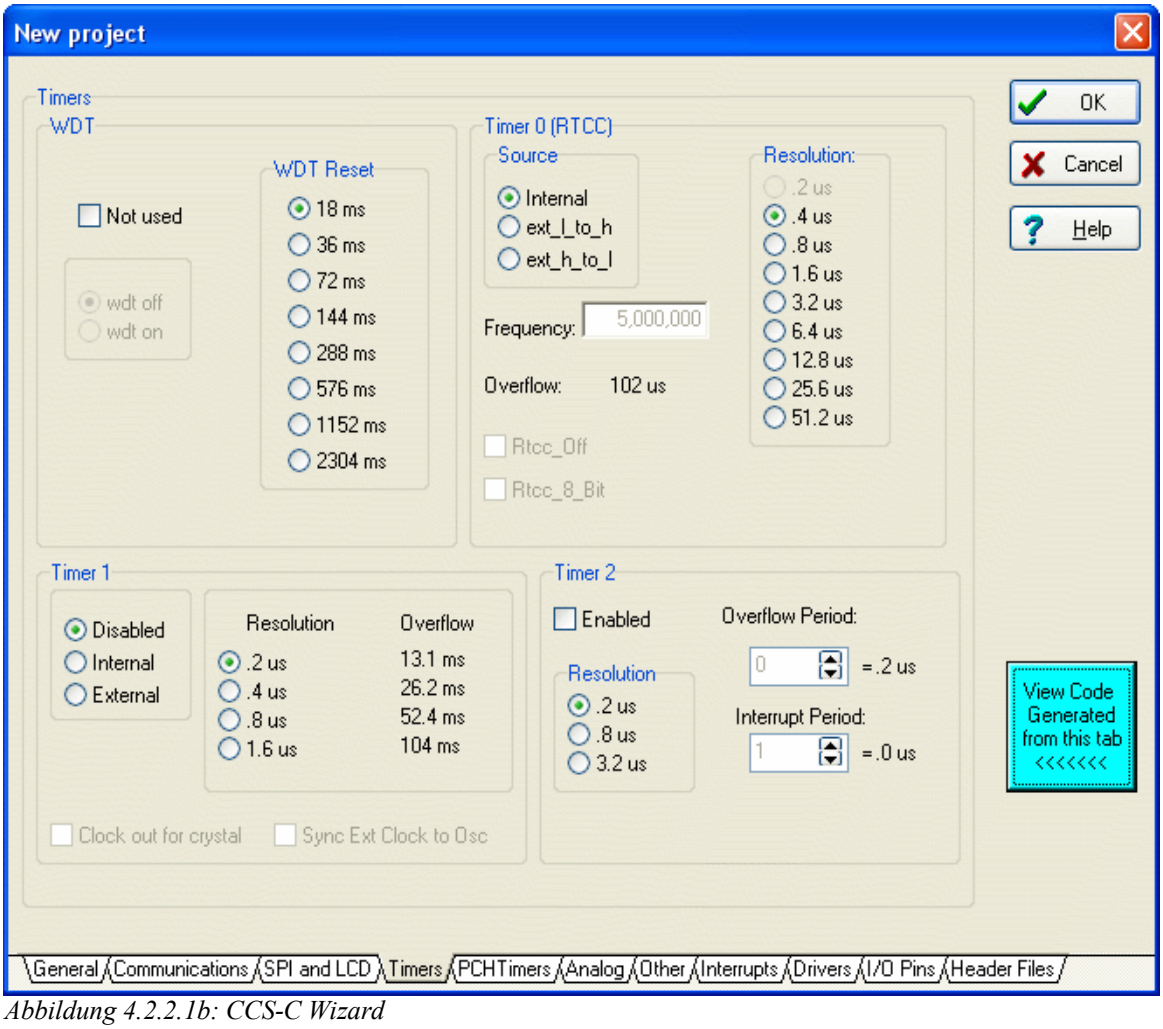

Abbildung 4.2.2.1b: CCS-C Wizard

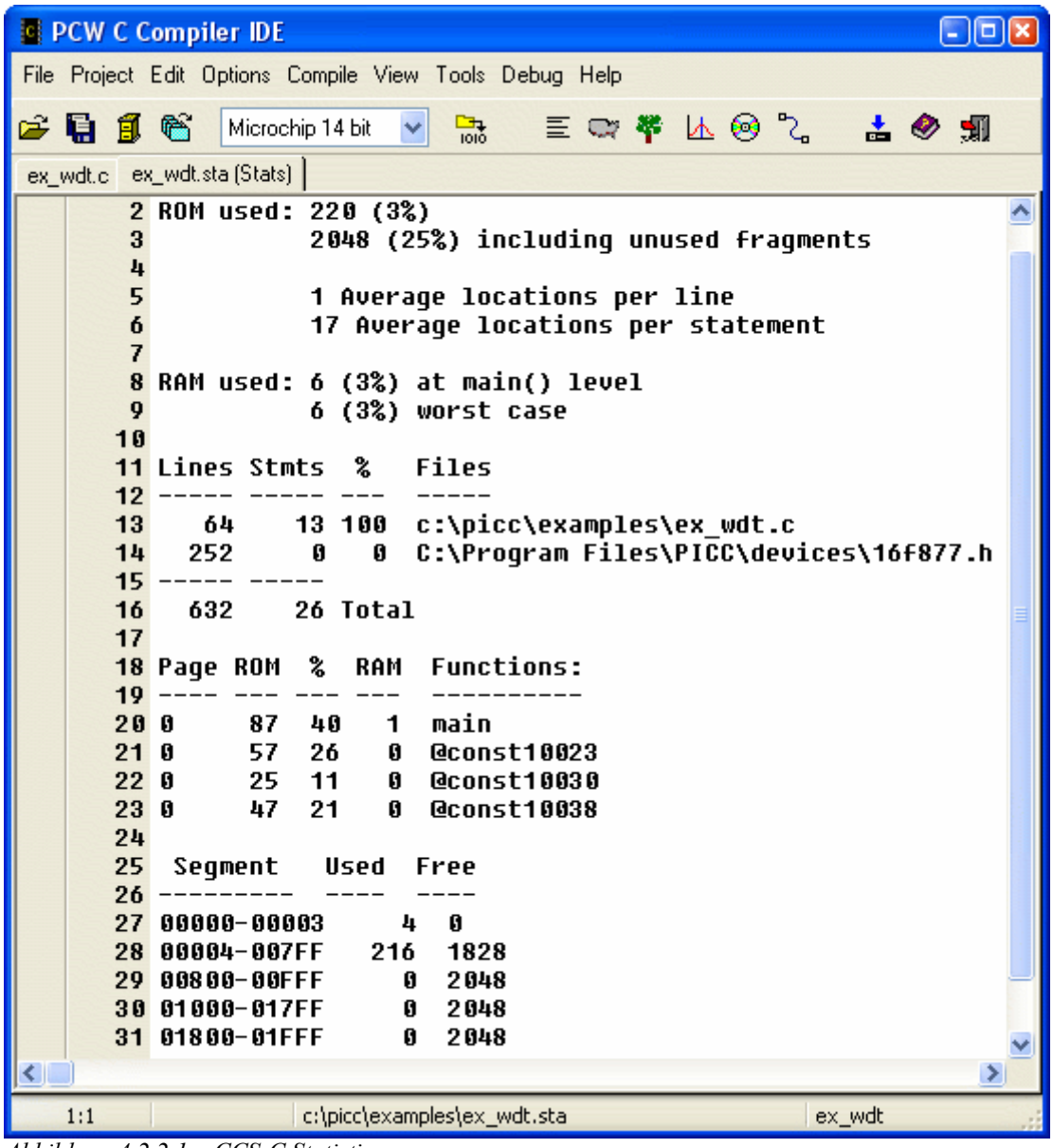

Abbildung 4.2.2.1c: CCS-C Statistic

4.2.2.2 Zustandsdiagramm

Hier ist nun ein vereinfachtes Zustandsdiagramm der „RFID-Watch-Box“ zu sehen (siehe Abb. 4.2.2.2a, Vollbilddarstellung im Anhang). Innerhalb der einzelnen Zustände sind wiederum Aktivitäten eingebettet, dessen Funktionalitäten jeweils in nebenstehenden Notizboxen beschrieben sind.

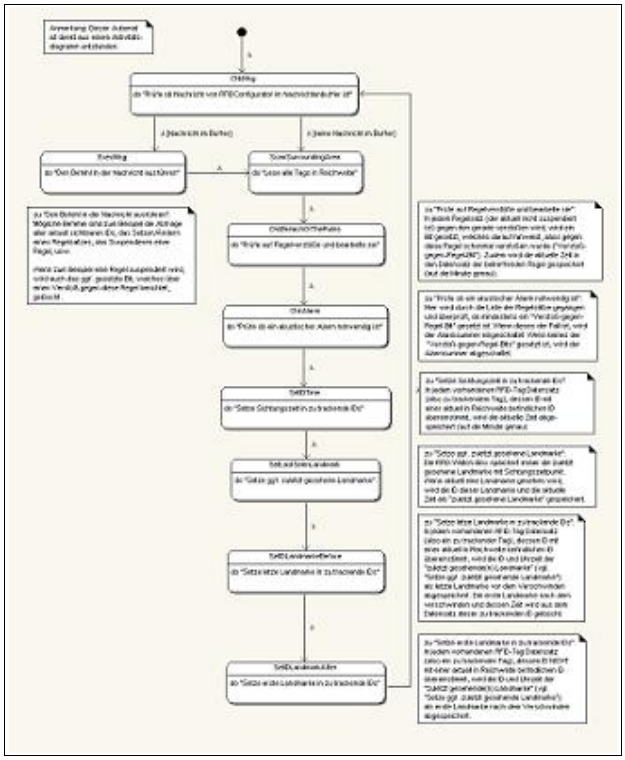

Abbildung 4.2.2.2a: Automat (miniaturisierte Ansicht, für das Vollbild siehe Anhang)

4.3 Protokoll zwischen „Management-Blackbox“ und PDA

Wie bereits im Kapitel 3.2.1 „Festlegung der Anforderungen“ beschrieben wurde, soll die Kommunikation zwischen der „RFID-Watch-Box“ und dem „RFID-Configurator“ auf Basis einer vom „RFID-Configurator“ ausgehenden pollenden Abfrage geschehen. Es wird angenommen, dass die „RFID-Watch-Box“ nur eine Anfrage zugleich abarbeiten kann. Wenn der „RFID-Configurator“ mehrere Anfragen hintereinander an die „RFID-Watch-Box“ sendet und eine Antwort zu einer dieser Anfragen erhält, muss der „RFID-Configurator“ annehmen, dass die restlichen Anfragen nicht empfangen wurden. Er muss die unbeantworteten Anfragen als neue Anfragen (mit neuer Nachrichtennummer) noch einmal stellen.

Die Nachrichten in beide Richtungen sollen folgendermaßen Aufgebaut sein:

SOH	Länge	Nummer	Nachrichtentyp	<< Nachricht >>	CRC-16

SOH (1 Byte):
Start einer Nachricht codiert mit 0x01 (ASCII: SOH - Start of Header)

Länge (2 Byte):
Nachrichtenlänge in Hexadezimal bestehend aus Nummer, Nachrichtentyp, Nachricht und CRC-16 in Bytes. Beispiel: Die Länge ist 0x36 Bytes: 0x33 0x36.

Nummer (4 Bytes):
Fortlaufende Nachrichtennummer in Hexadezimal. Antworten der „RFID-Watch-Box“ kommen mit der gleichen Nachrichtennummer, wie die Anfrage vom „RFID-Configurator“.

Nachrichtentyp (2 Bytes):
Anhand des Nachrichtentyps wird erkannt, wie die Datenstruktur von <<Nachricht>> aufgebaut ist. Der Nachrichtentyp mit der Nummer A7 würde zum Beispiel 0x41 0x37 codiert sein (ASCII).

<<Nachricht>> (x Bytes):
Die Nachrichten der unterschiedlichen Nachrichtentypen wird folgend beschrieben. Ihre Länge ist unterschiedlich.

CRC-16 (4 Bytes):
Checksumme über Nummer, Nachrichtentyp und Nachricht.
CRC-16 ergibt 2 Byte lange Checksummen.
Beispiel: Checksumme 0x7AD4 = 0x37 0x41 0x44 0x34

Nun werden die Nachrichtentypen und die Beschreibung ihrer Nachrichtenstrukturen nach folgendem Schema aufgeführt:

„Typnummer“: „Beschreibung (wie in den USE-Case-Diagrammen)“

„Struktur des Nachrichtenteils“

„ggf. kurze Erläuterung“

Nachrichten vom „RFID-Configurator“ zur „RFID-Watch-Box“:

00: Für zukünftige Anwendungen.

noch nicht definiert

01: „Anfrage, ob jetzt eine Anfrage möglich ist“

leer

11: „setze eine zu trackende RFID-Tag ID“

ID-Länge	ID	TypeOfID

ID-Länge (2 Bytes): Anzahl x der Länge der folgenden ID
ID (x Bytes): ID eines RFID-Tags
TypeOfID (1 Byte): Typ eines RFID-Tags, zum Beispiel Landmarke

12: „lese alle IDs von in Reichweite befindlichen Tags“

leer

13: „lese RFID-Tag Tracking-Daten eines Tags inklusive Typ“

ID-Länge	ID

ID-Länge (2 Bytes): Anzahl x der Länge der folgenden ID
ID (x Bytes): ID eines RFID-Tags

14: „lese alle IDs von zu trackenden (also bekannten) Tags“

ID-Länge	ID

ID-Länge (2 Bytes): Anzahl x der Länge der folgenden ID

ID (x Bytes): ID eines RFID-Tags

15: „lösche eine ID aus der Liste der zu trackenden RFID-Tags“

ID-Länge	ID

ID-Länge (2 Bytes): Anzahl x der Länge der folgenden ID

ID (x Bytes): ID eines RFID-Tags

21: „setze bzw. überschreibe eine Regel“

Regel Nr.	+/- Tag 1	ID-Länge1	ID 1	+/- Tag 2	ID-Länge2	ID 2

Regel-Nr. (3 Byte): Wenn zu setzende Regel die Nummer 2638 bzw. 0xA4E hat, lauten die drei Bytes 0x41, 0x34 und 0x45. Also auch hier ASCII-Kodierung

+/- Tag 1 (1 Byte): verlangte Anwesenheit von Tag 1

ID-Länge1 (2 Bytes): Anzahl x der Länge der folgenden ID 1

ID 1 (x Bytes): ID des RFID-Tags 1

Tag 2 (1 Byte): verlangte Anwesenheit von Tag 2

ID-Länge2 (2 Bytes): Anzahl y der Länge der folgenden ID 2

ID 2 (y Bytes): ID des RFID-Tags 1

Anmerkung: Suspendierungsende wird hier auf 0000-00-00-00-00 gesetzt, sodass die Regel auf jeden Fall nicht suspendiert ist.

22: „frage nach freier Regelnummer“

leer

23: „lese alle belegten Regelnummern“

leer

24: „lese Regeldaten einer Regel“

Regel-Nr.

Regel-Nr. (3 Bytes): Wenn zu lesende Regel die Nummer 2638 bzw. 0xA4E hat, lauten die drei Bytes 0x41, 0x34 und 0x45. Also auch hier ASCII-Kodierung

25: „lösche eine Regel“

Regel-Nr.

Regel-Nr. (3 Bytes): Wenn zu löschende Regel die Nummer 2638 bzw. 0xA4E hat, lauten die drei Bytes 0x41, 0x34 und 0x45. Also auch hier ASCII-Kodierung

31: „lese akuelle Regelverstöße“

leer

32: „setze Suspendierungszeitraum einer Regel“

Regel-Nr.	Jahr	Monat	Tag	Stunde	Minute

Regel-Nr. (3 Bytes): Wenn zu suspendierende (bzw. durch Nullen zu aktivierende) Regel die Nummer 2638 bzw. 0xA4E hat, lauten die drei Bytes 0x41, 0x34 und 0x45. Also auch hier ASCII-Kodierung

Jahr (4 Bytes): Jahr vom Ende der Suspendierung
(Beispiel: Jahr 2005 => 0x32 0x30 0x30 0x35)

Monat (2 Bytes): Monat vom Ende der Suspendierung
(Beispiel: März => 0x30 0x35)

Tag (2 Bytes): Tag vom Ende der Suspendierung
(Beispiel: 27.des Monats => 0x32 0x37)

Stunde (2 Bytes) Stunde vom Ende der Suspendierung
(Beispiel 21 Uhr => 0x32 0x31)

Minute (2 Bytes) Minute vom Ende der Suspendierung
(Beispiel 47. Minute => 0x34 0x37)

41: „setze Uhrzeit“

Jahr	Monat	Tag	Stunde	Minute	Sekunde

Jahr (4 Bytes):	Jahr (Beispiel: Jahr 2005 => 0x32 0x30 0x30 0x35)
Monat (2 Bytes):	Monat (Beispiel: März => 0x30 0x35)
Tag (2 Bytes):	Tag (Beispiel: 27.des Monats => 0x32 0x37)
Stunde (2 Bytes)	Stunde (Beispiel 21 Uhr => 0x32 0x31)
Minute (2 Bytes)	Minute (Beispiel 47. Minute => 0x34 0x37)
Sekunde (2 Bytes)	Sekunde (Beispiel 25. Sekunde => 0x32 0x35)
Anmerkung:	So wird die Uhrzeit (mit Datum) vom „RFID-Configurator“ in der „RFID-Watch-Box“ gesetzt.

42: „lese Uhrzeit“

leer

Anmerkung:	So ließt der „RFID-Configurator“ die aktuelle Uhrzeit (mit Datum) aus der „RFID-Watch-Box“.

Nachrichten von der „RFID-Watch-Box“ zum „RFID-Configurator“:

91: Antwort auf 01-Nachricht „Anfrage, ob jetzt eine Anfrage möglich ist“

JaOderNein

JaOderNein (1 Byte): 'T' (ASCII) für „True“, also „Ja, geht“
'F' (ASCII) für „False“, also „Nein, geht nicht“

A1: Antwort auf 11-Nachricht „setze eine zu trackende RFID-Tag ID“

HatGeklappt	Infotext

HatGeklappt (1 Byte): 'T'/'F' (ASCII)

Infotext (30 Bytes): erklärender ASCII-Text
(zum Beispiel: „Nein, geht im Moment nicht“)

A2: Antwort auf 12-Nachricht „lese alle IDs von in Reichweite befindlichen Tags“

Anzahl Tags	ID-Länge i	ID i	ID-Länge i+1	ID i+1	...

Anzahl Tags (3 Byte): Anzahl der folgenden Tags

ID-Länge i (2 Bytes): Anzahl x_i der Länge der folgenden ID i

ID i (x_i Bytes): ID des RFID-Tags i

ID-Länge i+1 (2 Bytes): Anzahl x_{i+1} der Länge der folgenden ID i+1

ID i+1 (x_{i+1} Bytes): ID des RFID-Tags i+1

...

Anmerkung: Maximale Anzahl ist durch die maximale Nachrichtenlänge begrenzt!

A3: Antwort auf 13-Nachricht „lese RFID-Tag Tracking-Daten eines Tags inklusive Typ“

TypOfID	TimeLastSeen	TimeOfLast LandmarkBefore LastSeen

TimeOfFirst LandmarkAfter LastSeen	ID-Länge von LastLandmark BeforeLastSeen	ID von LastLandmark BeforeLastSeen

ID-Länge von FirstLandmarkAfter LastSeen	ID von FirstLandmarkAfter LastSeen

TypeOfId (1 Byte):	Typ eines RFID-Tags
TimeLastSeen (12 Bytes):	Zeit der letzten Sichtung des RFID-Tags (beinhaltet Jahr, Monat, Tag, Stunde, Minute)
TimeOfLastLandmark BeforeLastSeen (12 Byte):	Zeit der letzten Sichtung einer Landmarke vor der letzten Sichtung des RFID-Tags.
TimeOfFirstLandmark AfterLastSeen:	Zeit der ersten Sichtung einer Landmarke nach der letzten Sichtung des RFID-Tags.
ID-Länge von LastLandmark BeforeLastSeen (2 Bytes):	ID-Länge x der letzten Landmarke vor der letzten Sichtung des RFID-Tags.
ID von LastLandmark BeforeLastSeen (x Bytes):	ID der letzten Landmarke vor der letzten Sichtung des RFID-Tags.
ID-Länge von FirstLandmark AfterLastSeen (2 Bytes):	ID-Länge y der ersten Landmarke nach der letzten Sichtung des RFID-Tags.
ID von FirstLandmark AfterLastSeen (y Bytes):	ID der ersten Landmarke nach der letzten Sichtung des RFID-Tags.

A4: Antwort auf 14-Nachricht „lese alle IDs von zu trackenden (also bekannten) Tags“

Anzahl Tags	ID-Länge i	ID i	ID-Länge i+1	ID i+1	...

Anzahl Tags (3 Byte): Anzahl der folgenden Tags

ID-Länge i (2 Bytes): Anzahl x_i der Länge der folgenden ID i

ID i (x_i Bytes): ID des RFID-Tags i

ID-Länge i+1 (2 Bytes): Anzahl x_{i+1} der Länge der folgenden ID i+1

ID i+1 (x_{i+1} Bytes): ID des RFID-Tags i+1

...

Anmerkung: Maximale Anzahl ist durch die maximale Nachrichtenlänge begrenzt!

A5: Antwort auf 15-Nachricht „lösche eine ID aus der Liste der zu trackenden RFID-Tags“

HatGeklappt	Infotext

HatGeklappt (1 Byte): 'T'/'F' (ASCII)

Infotext (30 Bytes): erklärender ASCII-Text (zum Beispiel: „Err, ID unbekannt“)

B1: Antwort auf 21-Nachricht „setze bzw. überschreibe eine Regel“

HatGeklappt	Infotext

HatGeklappt (1 Byte): 'T'/'F' (ASCII)

Infotext (30 Bytes): erklärender ASCII-Text (zum Beispiel: „Err, Regel-Nr. zu groß“)

B2: Antwort auf 22-Nachricht „frage nach freier Regelnummer“

HatFreieNummer	Regelnummer

HatFreieNummer (1 Byte): 'T'/'F' (ASCII)

Regelnummer (3 Bytes): HatFreieNummer = 'T': freie Regelnummer
HatFreieNummer = 'F': undefiniert

B3: Antwort auf 23-Nachricht „lese alle belegten Regelnummern“

Anzahl Regel-Nr.	Regel-Nr. i	Regel-Nr. i+1	Regel-Nr. i+2	Regel-Nr. i+3	...

Anzahl Regel-Nr. (3 Byte): Anzahl der folgenden Regeln

Regel-Nr. i (3 Bytes): Regelnummer i

Regel-Nr. i+1 (3 Bytes): Regelnummer i+1

...

Anmerkung: Max. Anzahl ist durch die max. Nachrichtenlänge begrenzt!

B4: Antwort auf 24-Nachricht „lese Regeldaten einer Regel“

+/- Tag 1	ID-Länge1	ID 1	+/- Tag 2	ID-Länge2	ID 2

Jahr	Monat	Tag	Stunde	Minute

+/- Tag 1 (1 Byte): verlangte Anwesenheit von Tag 1

ID-Länge1 (2 Bytes): Anzahl x der Länge der folgenden ID 1

ID 1 (x Bytes): ID des RFID-Tags 1

Tag 2 (1 Byte): verlangte Anwesenheit von Tag 2

ID-Länge2 (2 Bytes): Anzahl y der Länge der folgenden ID 2

ID 2 (y Bytes): ID des RFID-Tags 1

Jahr (4 Bytes):	Jahr der letzten Regelverletzung
Monat (2 Bytes):	Monat der letzten Regelverletzung
Tag (2 Bytes):	Tag der letzten Regelverletzung
Stunde (2 Bytes):	Stunde der letzten Regelverletzung
Minute (2 Bytes):	Minute der letzten Regelverletzung

B5: Antwort auf 25-Nachricht „lösche eine Regel“

HatGeklappt	Infotext

HatGeklappt (1 Byte):	'T'/'F' (ASCII)
Infotext (30 Bytes):	erklärender ASCII-Text (zum Beispiel: „Err, Regel-Nr. zu groß“)

C1: Antwort auf 31-Nachricht „lese akuelle Regelverstöße“

Anzahl	Regelnummer

Anzahl (1 Byte):	Anzahl aktueller Regelverstöße 0 -> keine Regelverstöße >8 -> mehr als acht Regelverstöße
Regelnummer (3 Bytes):	eine verstoßene Regel

Anmerkung: Als Antwort auf eine 31-Nachricht wird die Anzahl der aktuellen Regelverstöße und die Regelnummer von einer der Regeln, gegen die verstoßen wurde, zurückgegeben

C2: Antwort auf 32-Nachricht „setze Suspendierungszeitraum einer Regel“

HatGeklappt	Infotext

HatGeklappt (1 Byte): 'T'/'F' (ASCII)

Infotext (30 Bytes): erklärender ASCII-Text
(zum Beispiel: „Err, Regel-Nr. zu groß“)

D1: Antwort auf 41-Nachricht „setze Uhrzeit“

HatGeklappt	Infotext

HatGeklappt (1 Byte): 'T'/'F' (ASCII)

Infotext (30 Bytes): erklärender ASCII-Text
(zum Beispiel: „Err, es gibt keinen 35. Mai“)

D2: Antwort auf 42-Nachricht „lese Uhrzeit“

Jahr	Monat	Tag	Stunde	Minute	Sekunde

Jahr (4 Bytes): aktuelles Jahr in der RFID-Watch-Box

Monat (2 Bytes): aktueller Monat in der RFID-Watch-Box

Tag (2 Bytes): aktueller Tag in der RFID-Watch-Box

Stunde (2 Bytes): aktuelle Stunde in der RFID-Watch-Box

Minute (2 Bytes): aktuelle Minute in der RFID-Watch-Box

Sekunde (2 Bytes): aktuelle Sekunde in der RFID-Watch-Box

Kapitel 5

Resümee & Ausblick

In diesem Kapitel wird sich noch einmal rückblickend mit dem entstandenen Design und dessen Realisierung kurz auseinandergesetzt. Zudem wird beleuchtet, in wieweit die gesetzten Ziele erreicht wurden. Es werden die Stärken und Schwächen des Systems angesprochen, sowie schlussendlich noch einige Verbesserungs- und Erweiterungsvorschläge gemacht.

5.1 Erreichte Ziele

Das am Anfang der Arbeit gesetzte Hauptziel wurde erreicht. Das System ist in der Lage dem privaten Konsumenten ein eigenständiges Objekt-Tracking-System auf Basis von RFID an die Hand zu geben und das sogar zu moderaten Preisen. Es kann zum Einen dazu genutzt werden, Gegenstände des täglichen Lebens wieder zu finden, indem es einem die Uhrzeit und den ungefähren Ort der letzten Sichtung verrät und zum Anderen gestattet es dem Benutzer, Regeln aufzustellen, die dafür sorgen, dass man diese Gegenstände erst gar nicht verlieren, da man auf einen möglichen Verlust sofort hingewiesen wird. Des Weiteren wurde durch die flexible Form der Regeln zusätzlich erreicht, dass man diese so anlegen kann, dass sie einen auch daran erinnern können, dass man zum Beispiel einzelne Gegenstände nicht aus versehen in eine bestimmte Sperrzone mitnimmt bzw. aus einer dafür vorgesehenen und ausschließlichen Nutzungszone herausnimmt. So könnte die RFID-Watch-Box zum Beispiel daran erinnern, dass sie selber nicht in eine Ex-Schutz-Zone verbracht werden darf, da ihr Geräteaufbau nicht für die Benutzung in explosionsgefährdeten Bereichen ausgelegt ist.

Umgekehrt könnte sie jedoch auch daran erinnern, dass bestimmte, markierte Dokumente nicht aus einem besonders gesicherten Raum herausgetragen werden dürfen, da sie zum Beispiel der Geheimhaltung unterliegen.

Die Durchführung dieser Diplomarbeit stellte mich vor die Herausforderung, mich notwendigerweise auch mal mit den Möglichkeiten und Gefahren der RFID-Technologie zu befassen. Erforderlich war auch eine Auseinandersetzung mit anderen Trackingsystemen und ihren verwendeten Positionsbestimmungsverfahren, um einen Überblick über den Stand dieser Technik zu erhalten. Probleme bereitete mit unter die Komplexität der gesamten Aufgaben des „RFID-Trackers“, die aber vor allem durch das Erstellen der Use-Case-Diagramme greifbarer wurde. Eine weitere Schwierigkeit lag darin, dass Sun die Unterstützung von Java auf Microsoft Windows Mobile bzw. PocketPC Computern eher etwas stiefmütterlich behandelt. Im Bereich der mobilen Geräte liegt Suns Hauptaugenmerk eindeutig bei Mobiltelefonen und Java MIDP 2.0. Besonders die Portierung von Swing nach PersonalJava stellte eine echte Herausforderung dar. Selbst als Swing dann endlich zur Verfügung stand, blieb noch die Aufgabe bestehen, so zu programmieren, dass man auch akzeptable Reaktionszeiten in der Benutzerinteraktion vorzuweisen hatte. Die Bluetoothverbindung und die Einarbeitung in diese Technik bereitete glücklicherweise keine größeren Umstände, da durch das „Cable Replacement Protocol“ RFCOMM die Kommunikation wie bei einer herkömmlichen RS232-Verbindung gehandhabt werden konnte. Mehr Möglichkeiten hätten natürlich bestanden, wenn ein direkter Zugriff auf den Bluetoothstack aus Java heraus möglich gewesen wäre, jedoch hat der Weg über die (nicht von Sun bereitgestellte) JavaComm-Bibliothek keine Einbußen bedeutet, die wirklich ausschlaggebend gewesen wären. Zudem wäre auch die Möglichkeit einer späteren Portierung auf eine MIDP 2.0 Mobiltelefon durch einen direkten Zugriff auf den Bluetoothstack erschwert worden, da unter MIDP 2.0 wiederum über RFCOMM kommuniziert werden müsste. Sehr ärgerlich empfand ich, dass das RFID-Lesegerät der Firma Megaset keinen Verpolungsschutz aufwies – ein klarer Designfehler! So kam es dazu, dass das mitgelieferte Netzteil wohl von einem anderen Studenten im Labor für seine Zwecke gebraucht und anschließend wieder mit dem Lesegerät verbunden wurde, ohne dabei jedoch auf die richtige Polung zu achten. Das Resultat war, dass ein Kondensator und der DC-DC-Wandler weggebrutzelt waren, sodass ich das Lesegerät erstmal reparieren musste. Wenn die Bearbeitungszeit für die Diplomarbeit noch etwas länger gewesen wäre, hätte ich die erste und aus Zeitgründen nicht in allen Funktionen vollständige Implementierung des Prototypen gerne noch komplettiert. Da jedoch in dieser Diplomarbeit die Hauptaufgabe in der Systemanalyse und dem Design eines RFID-Trackingsystems liegt, dürfte der implementierte Proof-Of-Concept ohnehin mehr als ein schönes Extra denn als unbedingt erforderlicher Teil der Aufgabe angesehen werden.

5.2 Ausblick

Nun folgend wird beschrieben, welche Verbesserungsmöglichkeiten oder gar Ergänzungen aus heutiger Sicht möglich und gegebenenfalls sogar wünschenswert sein könnten.

5.2.1 Das Problem der Lesegeräte-Kollision

Leider existieren bis Heute nur für den Fall von mehreren gleichzeitig antwortenden RFID-Tags standardisierte Antikollisionsverfahren. Einen derartigen Weg für die Kollisionerkennung oder gar Kollisionsvermeidung von Lesegeräten gibt es bisher nicht. Genau betrachtet, gibt es sogar zwei Arten von Kollisionen, welche durch Lesegeräte ausgelöst werden können. Zum einen kann es vorkommen, dass zwei Lesegeräte gleichzeitig versuchen mit den selben RFID-Tags in Kontakt zu treten. Wenn sich nun beide Lesegeräte dabei gleichzeitig im relativen Nahbereich eines RFID-Tags befinden, kann dieses Tag weder das eine noch das andere Lesegerät verstehen, da es zu starken Interferenzen zwischen den beiden Lesegeräten kommt. Die zweite Arte der Kollision besteht darin, dass ein RFID-Tag gerade dabei ist, einem in seinem Nahbereich befindlichen Lesegerät zu antworten. Die Signalstärke des RFID-Tags ist relativ schwach, sodass ein zweites RFID-Lesegerät, welches mit einer sehr viel höheren Signalstärke arbeitet, diese Kommunikation stören kann, auch wenn die Entfernung zwischen erstem und zweitem Lesegerät sehr viel größer ist, als die Entfernung zwischen erstem Lesegerät und dem gerade antwortenden RFID-Tag. Zumindest für die zweite Arte der Kollision gibt es bereits eine Lösung, indem die Energieversorgung des RFID-Tags und die Kommunikation von Lesegerät zu RFID-Tag auf einer vollkommen anderen Frequenz stattfindet, als die signalschwache Antwort des RFID-Tags zum Lesegerät. Ein Beispiel für ein System mit anharmonischer Rückfrequenz ist die Eurobalise des europäischen Bahnverkehrs. Hier erfolgt die Energieversorgung der Balise[1] auf einer Frequenz von 27,115 MHz und die Kommunikation von Balise zum Triebfahrzeug findet auf 4,24 MHz statt. Aber auch schon die Verwendung eines Hilfsträgers innerhalb der Lastmodulation hilft in gewissen Grenzen einer Störung durch ein zweites Lesegerät vorzubeugen. Da im Falle des in dieser Diplomarbeit entwickelten Trackingsystems das Lesegerät ständig seine Position verändert und dabei

Abbildung 5.2.1a: Eurobalise auf Gleisanlage

1 Balise ist das französische Wort für Bake

unentwegt Funksignale aussendet, kann es natürlich sehr leicht zu gegenseitigen Störungen mit anderen RFID-Systemen kommen, die ebenfalls auf 13,56 MHz arbeiten. Darum wäre es nicht nur für dieses hier vorliegende System, sondern auch ganz allgemein sehr wünschenswert, wenn es dafür zukünftig eine generelle Lösung geben würde.

5.2.2 Reichweitenproblematik

Das im Prototyp verwendete RFID-Lesegerät hat nur eine sehr geringe Reichweite von ca. 10 Zentimetern. Tatsächlich erforderlich sind jedoch etwa zwei Meter. In Europa sind bisher nur Lesegeräte bis 0,5 Watt erlaubt, wobei eine Reichweite zu passiven Tags von maximal 120 Zentimetern erreicht wird. Das UCC[1] versucht aber auf das ETSI[2] daraufhin einzuwirken, dass bald auch Lesegeräte mit bis zu zwei Watt zugelassen werden. Der Vorschlag sieht als Strategie zur Kollisionsvermeidung zwischen Lesegeräten ein „Carrier Sense“ vor, sodass diese, wenn ein Frequenz-Kanal durch ein anderes Lesegerät belegten ist, auf einen anderen von insgesamt zehn Kanälen wechseln können (RFID Journal, [49]).

5.2.3 Notwendigkeit von sich authentifizierenden RFID-Tags

Um Täuschungsversuche durch Angreifer abzuwehren, welche zum Beispiel das Entfernen eines durch ein RFID-Tag markierten Gegenstand verschleiern wollen oder aber das System derart Manipulieren wollen, dass durch falsch gesetzte RFID-Landmarken eine Orientierung unmöglich wird, sollten authentifizierende RFID-Tags verwendet werden.

5.2.4 Notwendigkeit von anonymen RFID-Tags

Sollte man auf die Idee kommen das System kommerziell nutzen zu wollen, könnte es auch interessant sein, ausgebrachte Landmarken, gegenüber nicht zahlenden Kunden oder Mitbewerbern anonym zu halten. Ein Beispiel dafür wären Landmarken in Tourismuszentren, mit deren Hilfe auf Sehenswürdigkeiten hingewiesen werden soll, die sich zum Beispiel innerhalb von Gebäuden oder Höhlen befinden, wo ein GPS basiertes System nicht greifen würde. Aber auch das Wiedererkennen von bei sich getragenen Gegenständen sollte möglichst durch anonyme RFID-Tags vermieden werden, um die Vertraulichkeit des Aufenthaltsortes der tragenden Person zu wahren.

1 Uniform Code Council
2 European Telecommunications Standards Institute

5.2.5 Der spezielle Objekt-Typ „Mensch“

Im Rahmen der Analyse wurde neben den RFID-Tag-Typen „Landmarke“ und „Objekt“ auch noch der Typ „Mensch“ eingeführt. Warum? Der Grund liegt in zwei möglichen Erweiterungen für das RFID-Trackingsystem. Zum einen könnte sich das System beim Verschwinden eines Objektes nicht nur die vor und nach dem Verschwinden auftauchenden Landmarken merken, sondern auch unmittelbar beim Verschwinden anwesende Personen. Mit dieser Information könnte sich der später suchende Benutzer des Systems gegebenenfalls wieder daran erinnern, dass er den verschwundenen Gegenstand womöglich auch an die damals automatisch gespeicherte Person verliehen hat. Eine zweite Erweiterungsmöglichkeit, die durch die Einführung eines Tag-Typen „Mensch“ (welcher dann an die RFID-Watch-Box zu koppeln wäre) möglich wird, ist, dass zwei oder mehr Benutzer von RFID-Watch-Boxen sich gegenseitig erkennen können, sodass die Watch-Boxen automatisch ihre Sichtungen von Gegenständen gegenseitig abgleichen. Das natürlich nur, wenn sich die Benutzer gegenseitig vertrauen. Dieses Abgleichen und Aktualisieren der Sichtungsdaten zwischen mehreren RFID-Watch-Boxen würde dazu führen, dass je mehr Watch-Boxen unterwegs sind, auch der Aufenthaltsort von verschwundenen Gegenständen immer aktueller gehalten werden kann – bis hin zu einer quasi flächendeckenden Überwachung.

5.2.6 Genauere Positionsbestimmung

Das erarbeitete System erlaubt es nur an bestimmten Punkten - den Landmarken - genaue Aussagen über den aktuellen Aufenthaltsort zu machen. Zwischen den Sichtungen von zwei Landmarken ist keine verlässliche Angabe möglich.

Wünschenswert wäre es, wenn die Landmarken nur Kalibrierungspunkte für ein anderes Positionsbestimmungssystem wären. In Gegenden, wo sich keine RFID-Landmarken befinden, jedoch ein Empfang von GPS-Signalen möglich ist, könnte auch eine vereinzelte GPS-Positionsbestimmung diesen Kalibrierungspunkt liefern. Nur wie lässt sich eine stetige Nachführung bei Ortswechseln durchführen? Bei Kraftfahrzeugen bedient man sich der Informationen, die man aus Tachogeschwindigkeit und Lenkradeinschlag beziehen kann. Doch leider bieten Menschen, die eine RFID-Watch-Box mit sich führen, keine solchen Messgrößen. Ein erster kleiner Ansatz wäre ein Schrittzähler, der eine grobe Abschätzung über die maximale Entfernung zum letzten Kalibrierungspunkt machen könnte - zumindest wenn sich der Benutzer nur per pedes fortbewegt. Auf jeden Fall wäre die Suche nach einem geeigneten Positionsnachführungssystem ein Interessanter Ansatzpunkt für weitere Forschungs- und Entwicklungsarbeit zur Erweiterung dieses RFID-Trackingsystems.

Hinweis: Die im Anhang erwähnten Internetseiten sind überwiegend in der digitalen Internet-Bibliothek http://www.archive.org archiviert. Sollte die ursprüngliche Seite zu Ungunsten verändert oder gänzlich entfernt worden sein, sollte auf alle Fälle versucht werden, über http://www.archive.org auf die archivierten Internetseiten zuzugreifen.

A. Abbildungsverzeichnis

2.3f: UKW-Drehfunkfeuer Hamburg auf 113.10 MHz
Quelle: http://www.tf.uni-kiel.de/~fp/fliegerei/ausbildung/
sprechfunk/grafiken/vor3a.gif
Zugriffsdatum: 2.02.2005

2.3g: Monitorsoftware
Quelle: http://www.nobbi.com/monitor/main_ctl.png
Zugriffsdatum: 29.01.2005

2.3h: Monitorsoftware
Quelle: http://www.nobbi.com/monitor/m20.png
Zugriffsdatum: 1.02.2005

2.3i: Funkzellen
Quelle: Ausschnitt aus
http://www.medienengineering.de/lehre/NWBasis/
AufgabenundThemen/Themen/Vortraege/
Lokalisationssysteme-Ryll.Tauchnitz.pdf
Zugriffsdatum: 29.01.2005

2.3j: Netmonitor eines Nokia Mobiltelefons
Quelle: http://www.nobbi.com/display/nokia11.jpg
Zugriffsdatum: 29.01.2005

2.3k: Cricket-Hardware vom MIT
Quelle: http://nms.lcs.mit.edu/projects/cricket/pictures/cricketv2.jpg
Zugriffsdatum: 1.02.2005

2.4a: Bluetooth Specification Protocol Stack
Quelle: http://www.palowireless.com/infotooth/
images/tutorial_images/spec_stack.gif
Zugriffsdatum: 1.02.2005

3.1.1a: Verbindung Name-Gegenstand
Quelle: Martin Stein

3.1.2a: Use-Case-Diagramm „Meldung von und Umgang mit aktuellen Regelverstößen“
Quelle: Martin Stein

3.1.2b: Use-Case-Diagramm „Verwaltung von RFID-Tags“
Quelle: Martin Stein

3.2.2b: Use-Case-Diagramm „Verwaltung von RFID-Tags“
Quelle: Martin Stein

3.2.2c: Use-Case-Diagramm „Verwaltung von Regeln“
Quelle: Martin Stein

3.3.3a: Normale Reichweite
Quelle: Martin Stein

3.3.3b: Erweiterte Reichweite mit zwischengeschaltetem Repeater
Quelle: Martin Stein

3.3.5a: TDOA-Antenne
Quelle: http://home.att.net/~jleggio/projects/rdf/555tdoa1.jpg
Zugriffsdatum: 1.02.2005

3.3.5b: Richtungsachse
Quelle: Ausschnitt aus http://members.aol.com/homingin/hfpatterns.gif
Zugriffsdatum: 1.02.2005

3.3.5c: Triangulation
Quelle: Martin Stein

3.3.5d: Richtung
Quelle: Ausschnitt aus http://members.aol.com/homingin/hfpatterns.gif
Zugriffsdatum: 1.02.2005

3.3.5e: Dopplerantenne
Quelle: http://www.dopsys.com/images/df_ant.jpg
Zugriffsdatum: 1.02.2005

3.3.5f: Dopplerantenne
Quelle: http://www.dopsys.com/images/ser6000/P5030001A.JPG
Zugriffsdatum: 1.02.2005

3.3.5g: Trackingsoftware für mobile Fahrzeugempfänger
Quelle: http://www.dopsys.com/images/ser6100/autodem.gif
Zugriffsdatum: 1.02.2005

3.3.5h: Professioneller Funkpeiler von Rohde&Schwarz
Quelle: http://www.rohde-schwarz.com/www/products_files.nsf/
img/ddf0xeimg.jpg/$file/ddf0xeimg.jpg
Zugriffsdatum: 1.02.2005

3.3.5i: Portabler Funkpeiler von Rohde&Schwarz
Quelle: http://www.rohde-schwarz.com/www/products_files.nsf/
img/119311img.jpg/$file/119311img.jpg
Zugriffsdatum: 1.02.2005

3.3.5j: Verdeckte Doppler-Antennen auf PKW
Quelle: http://www.dopsys.com/images/ser6100/6073x95.jpg
Zugriffsdatum: 1.02.2005

3.3.5k: Selbstbau Doppler-Peiler
Quelle: http://members.tripod.com/~clearRX/SDFFOTO.JPG
Zugriffsdatum: 1.02.2005

3.3.6a: Ein Ausschnitt aus der RFID-Watch-Box Hardware
Quelle: Martin Stein

4.1a: Übersicht
Quelle: Martin Stein

4.1.1a: ISO-Reader-Controller
Quelle: Ausschnitt aus http://www.megaset.com/Produkte/Gewaehrl_RFID/
ProdukteRFID/RFID_1356MHz_ISO-Reader_Dok_300.pdf
Zugriffsdatum: 26.01.2005

4.1.1b: ISO-Reader-Box
Quelle: Ausschnitt aus http://www.megaset.com/Starter_Kits/
ISO-Reader-Box_800x600.jpg
Zugriffsdatum: 26.01.2005

4.1.2a: Schaltung der Management-Blackbox
Quelle: Martin Stein

4.1.2b: 16F877A Pinout
Quelle: siehe 4.1.2c

4.1.2c: 16F877A (PDIP)
Quelle: Ausschnitt aus http://ww1.microchip.com/downloads
/en/DeviceDoc/39582b.pdf
Zugriffsdatum: 26.01.2005

4.1.2d: 24AA512 Pinout
Quelle: siehe 4.1.2e

4.1.2e: 24AA512 Block Diagram
Quelle: Ausschnitt aus http://ww1.microchip.com/downloads
/en/DeviceDoc/21754E.pdf
Zugriffsdatum: 26.01.2005

4.1.2f: MAX233 Pinout und typische Beschaltung
Quelle: Ausschnitt aus http://pdfserv.maxim-ic.com/en/ds/MAX220-MAX249.pdf
Zugriffsdatum: 26.01.2005

4.1.2g: Pico Core
Quelle: Ausschnitt aus http://www.sphinx-electronics.de/deutsch
/bluetooth/downloads/DB_PICOCore_240402.pdf
Zugriffsdatum: 26.01.2005

4.1.2h: Pico Plug
Quelle: http://www.sphinx-electronics.de
Zugriffsdatum: 26.01.2005

4.1.4a: PDA mit RFID-Configurator Software
Quelle: Martin Stein

4.2.1.2a: Freie Stilisierung der Zweiteilung von GUI und Management-Logik
Quelle: Martin Stein

4.2.1.2b: Klassendiagramm (miniaturisierte Ansicht – Vollbild siehe Anhang)
Quelle: Martin Stein

4.2.1.3a: Sequenzdiagramm „Abfrage der aktuellen Regelverstöße“
Quelle: Martin Stein

4.2.1.3b: Sequenzdiagramm „Akzeptieren eines aktuellen Regelverstoßes“
Quelle: Martin Stein

4.2.1.3c: „Tags In Range“
Quelle: Martin Stein

4.2.1.3d: Sequenzdiagramm „Abfrage von Tags In Range“
Quelle: Martin Stein

4.2.1.3e: Sequenzdiagramm „Abfrage der genauen Daten eines einzelnen RFID-Tags“
Quelle: Martin Stein

4.2.1.3f: Sequenzdiagramm „speichern eines RFID-Tags“
Quelle: Martin Stein

4.2.1.3g: Sequenzdiagramm „Updaten der Tags in Reichweite“
Quelle: Martin Stein

4.2.1.3h: jTabbedPane leer
Quelle: Martin Stein

4.2.1.3i: jTabbedPane geladen
Quelle: Martin Stein

4.2.1.3j: Sequenzdiagramm „Nachladen des RulePanels“
Quelle: Martin Stein

4.2.2.1a: CCS-C IDE
Quelle: http://www.ccsinfo.com/pix/PCW.gif
Zugriffsdatum: 26.01.2005

4.2.2.1b: CCS-C Wizard
Quelle: http://www.ccsinfo.com/pix/Wizard_Timers.gif
Zugriffsdatum: 26.01.2005

4.2.2.1c: CCS-C Statistic
Quelle: http://www.ccsinfo.com/pix/Statistics.gif
Zugriffsdatum: 26.01.2005

4.2.2.2a: Automat (miniaturisierte Ansicht, für das Vollbild siehe auch Anhang)
Quelle: Martin Stein

5.2.1a: Eurobalise auf Gleisanlage
Quelle: Ausschnitt aus http://references.transportation.siemens.com/refdb
/link_download.jsp?file_name=A19100-V100-B846607.pdf&l=de
Zugriffsdatum: 11.02.2005

B. Literaturverzeichnis

[1] METRO Group: *METRO Group, Wal-Mart und Tesco treiben Thema RFID voran*
http://www.metrogroup.de/servlet/PB/menu/1012074_l1/index.html
Zugriffsdatum: 15.02.2005

[2] Niccolai, James (Computerworld): *DaimlerChrysler sees savings in RFID tags*
http://www.computerworld.com/mobiletopics/mobile
/technology/story/0,10801,91307,00.html
Zugriffsdatum: 15.02.2005

[3] Sullivan, Laurie (InformationWeek): *Logistics Providers Ready RFID Services*
http://www.informationweek.com/showArticle.jhtml?articleID=22102220
Zugriffsdatum: 12.02.2005

[4] Coates, Ron (CNET News.com): *At Delta, tracking bags with radio tags*
http://news.zdnet.com/2100-1009_22-5254118.html
Zugriffsdatum: 13.02.2005

[5] US-Navy: *Expeditionary Logistics and Combat Service Support*
http://www.nfesc.navy.mil/amphib/products/line2/default.asp
Zugriffsdatum: 14.02.2005

[6] Finkenzeller, Klaus: *RFID-Handbuch 3. Auflage*, Seite 30 ff, ISBN 3446220712

[7] Finkenzeller, Klaus: *RFID-Handbuch 3. Auflage*, Seite 38 ff, ISBN: 3446220712

[8] Multipick-Service – Deutschland: *Online-Shop*
http://www.multipick-service.de
Zugriffsdatum: 14.02.2005

[9] Finkenzeller, Klaus: *RFID-Handbuch 3. Auflage*, ISBN 3446220712
(http://rfid-handbook.de)

[10] Pro Tech Monitoring: *Advanced Systems that offer the Ultimate View*
http://www.ptm.com
Zugriffsdatum: 14.02.2005

[11] Vermessungsverwaltungen der Länder: *SAPOS*
http://www.sapos.de
Zugriffsdatum: 14.02.2005

[12] esp@cenet database: *europäisches Patent EP0847537*
http://l2.espacenet.com/espacenet/viewer?PN=EP0847537
Zugriffsdatum: 14.02.2005

[13] delphion (kostenpflichtig): *US-Patent 6,018,313*
http://www.delphion.com/details?pn=US06018313__
Zugriffsdatum: 14.02.2005

[14] United States Patent and Trademark Office (kostenfrei): *US-Patent 6,018,313*
http://patft.uspto.gov/netahtml/srchnum.htm
(kein direkter Link: nach „6,018,313“ suchen)
Zugriffsdatum: 14.02.2005

[15] Busch, H.: *Funk-Navigation: LORAN*
http://www.seefunknetz.de/loran.htm
Zugriffsdatum: 14.02.2005

[16] wikipedia (DE): *Funkfeuer*
http://de.wikipedia.org/wiki/Funkfeuer_%28Seefahrt%29
Zugriffsdatum: 14.02.2005

[17] wikipedia (EN): *VHF omnidirectional range*
http://en.wikipedia.org/wiki/VOR
Zugriffsdatum: 14.02.2005

[18] Cricket-Project (MIT): *The Cricket Indoor Location System*
http://nms.lcs.mit.edu/projects/cricket/
Zugriffsdatum: 14.02.2005

[19] Projekt „MoMa - Mobiles Marketing“: *Vergleich unterschiedlicher Ortungstechnologien unter Berücksichtigung der Kosten-/Nutzenaspekte*
http://www.mobilesmarketing.com/moma/Downloads
/Trendberichte/MoMa_PosPapier_Ortung.pdf
Zugriffsdatum: 14.02.2005

[20] Siemens VDO Automotive: *Ausrüstung entlang der Strecke...*
...Vorfahrt für Bus und Strassenbahn
http://www.siemens-tts.ch/haupt.asp?nv=2142&spr=1
Zugriffsdatum: 14.02.2005

[21] Hamburger Abendblatt: *So funktioniert die Busbeschleunigung*
http://www.abendblatt.de/daten/2004/05/27/299327.html
Zugriffsdatum: 14.02.2005

[22] Siemens VDO Automotive: *Referenzen Betriebsleitsysteme*
http://www.siemens-tts.ch/file/1/ct_100_ctDownload1.pdf
Zugriffsdatum: 14.02.2005

[23] palowireless: *K5 - Serial Port Profile* (siehe: Kapitel 5.3.1)
http://www.palowireless.com/infotooth/tutorial/k5_spp.asp
Zugriffsdatum: 14.02.2005

[24] palowireless: *RFCOMM Protocol*
http://www.palowireless.com/infotooth/tutorial/rfcomm.asp
Zugriffsdatum: 14.02.2005

[25] Hennig, Ladkin, Sieker (RVS Group, Universität Bielefeld):
Privacy Enhancing Technology Concepts for RFID Technology Scrutinised
http://www.rvs.uni-bielefeld.de/publications/Reports
/SPC2005_Privacy_Enhancing_Technology_Concepts
_for_RFID_Technology_Scrutinised.pdf
Zugriffsdatum: 14.02.2005

[26] BSI: *Risiken und Chancen des Einsatzes von RFID-Systemen*
http://www.bsi.bund.de/fachthem/rfid/RIKCHA.pdf
Zugriffsdatum: 14.02.2005

[27] Langheinrich, Marc: *Die Privatsphäre im Ubiquitous Computing - Datenschutzaspekte der RFID-Technologie*
http://www.vs.inf.ethz.ch/publ/papers/langhein2004rfid.pdf
Zugriffsdatum: 14.02.2005

[28] Fraunhofer IPA: *Laufzeitmessung von RFID-Signalen zur ortsauflösenden Objektlokalisierung*
http://www.ipa.fhg.de/Stellenmarkt/showjob.php?Nr=1423
Zugriffsdatum: 14.02.2005

[29] inside-it: Hilfe! *Der implantierbare RFID-Chip kommt*
http://www.inside-it.ch/frontend/insideit?XE7lhitk44WlAAgv
8MWOSSsZamkEpxsG2fV0sZW66WaXAef1BbqKV7ovoE8j
Zugriffsdatum: 14.02.2005

[30] Kushner, David (Übersetzung: Schwan, Ben): *RFID-Technik geht unter der Haut*
http://www.heise.de/tr/aktuell/meldung/53785
Zugriffsdatum: 14.02.2005

[31] DBC Braincon Technologies GmbH: *LEGIC RFID ZUTRITTSKONTROLLE*
http://www.dbcbraincon.de/600.htm
Zugriffsdatum: 14.02.2005

[32] DBC Braincon Technologies GmbH:
LEGIC RFID SCHRANKVERRIEGELUNG GAT LOCK 5000
http://www.dbcbraincon.de/schrank.htm
Zugriffsdatum: 14.02.2005

[33] 3S design GmbH: *3S Xccess Zugangskontroll-System*
http://zugang.3sdesign.de/index.htm?lesemodulfeatures.htm
Zugriffsdatum: 14.02.2005

[34] United States Patent and Trademark Office (kostenfrei): *US-Patent 5,758,277*
http://patft.uspto.gov/netahtml/srchnum.htm
(kein direkter Link: nach „5,758,277“ suchen)
Zugriffsdatum: 14.02.2005

[35] United States Patent and Trademark Office (kostenfrei): *US-Patent 5,005,210*
http://patft.uspto.gov/netahtml/srchnum.htm
(kein direkter Link: nach „5,005,210“ suchen)
Zugriffsdatum: 14.02.2005

[36] Motron Electronics: *TxID Transmitter FingerPrinter*
http://www.motron.com/TransmitterID.html
Zugriffsdatum: 14.02.2005

[37] Rager, Richard: *Xmit_ID*
http://xmit.penguinman.com/xmit_id.html
Zugriffsdatum: 14.02.2005

[38] Leggio, Joe: *555 Time-Difference-Of-Arrival RDF*
http://home.att.net/~jleggio/projects/rdf/tdoa2.htm
Zugriffsdatum: 14.02.2005

[39] Tiedemann, Klaus: *Wie baut man einen Doppler-Peiler?*
http://members.tripod.com/~clearRX/RDFUNITD.HTM
Zugriffsdatum: 14.02.2005

[40] Lythall, Harry: *DIRECTION FINDING SYSTEM*
http://web.telia.com/~u85920178/rx/df-00.htm
Zugriffsdatum: 14.02.2005

[41] Kuhn, Markus G.:
Compromising emanations: eavesdropping risks of computer displays
http://www.cl.cam.ac.uk/TechReports/UCAM-CL-TR-577.pdf
Zugriffsdatum: 14.02.2005

[42] Zivadinovic, Dusan: *Firstclass Luftverkehr - Bluetooth setzt zum Boom an*
http://www.heise.de/ct/03/23/142/
Zugriffsdatum: 14.02.2005

[43] blueboard (Verfasser: „US101“):
Lurker's Guide to Running PersonalJava on WinCE
http://www.blueboard.com/j2me/notes/2002_7_26.htm
Zugriffsdatum: 14.02.2005

[44] BUGPARADE REPORT: *Swing 1.1.1 uses a faulty test for JVM version PJAVA*
http://adorphuye.com/4309057/4309057.html
Zugriffsdatum: 14.02.2005

[45] „teilo“: *CEJavaComm*
http://www.teilo.net
Zugriffsdatum: 14.02.2005

[46] RXTX: *RXTX*
http://www.rxtx.org/
Zugriffsdatum: 14.02.2005

[47] Hobot, Michal: *RXTX for iPAQ*
http://republika.pl/mho/java/comm/RXTX_iPAQ_0211.zip
Zugriffsdatum: 14.02.2005

[48] MEGASET Systemtechnik: *Dokumentation ISO-Reader* (Seite 25)
http://museum.foebud.org/texte/aktion/rfid/
RFID%2013,56MHz%20ISO-Reader_Dok.pdf
Zugriffsdatum: 14.02.2005

[49] Finke, Thomas und Kelter, Harald (BSI):
Abhörmöglichkeiten der Kommunikation zwischen Lesegerät und Transponder am Beispiel eines ISO14443-Systems
http://www.bsi.de/fachthem/rfid/Abh_RFID.pdf
Zugriffsdatum: 18.02.2005

[50] c't: *Blooover demonstriert schwere Sicherheitslücken bei Bluetooth-Handys*
http://www.heise.de/security/news/meldung/print/54659
Zugriffsdatum: 18.02.2005

C. Weiterführende Bilder, Dokumente und Patente

Auf den nächsten Seiten sind folgende Anhänge zu finden:

1.) Klassendiagramm des „RFID-Configurators“

2.) Zustandsdiagramm der „RFID-Watch-Box“

3.) RASANT-Patent, europäisches Patentamt EP0847537 (Quelle: [12])

4.) RASANT-Patent, US-Patent 6,018,313 (Quelle: [14])

5.) US-Patent 5,758,277 (Quelle: [34])

6.) US-Patent 5,005,210 (Quelle: [35])

Hinweis:

Der folgende Anhang befindet sich ausschließlich in der gedruckten Fassung der Diplomarbeit. Die gedruckte Fassung kann in der „Bibliothek II“ (Berliner Tor 7, 20099 Hamburg) ausgeliehen werden.