Universität Karlsruhe (TH), Fakultät für Mathematik,
Institut für Mathematische Stochastik

Verteilung von Ausgabeströmen in Ubiquitous Computing Umgebungen

Diplomarbeit

von

M. Sc., Cand.-Math. techn.
Andreas Neumann

Tag der Anmeldung : 1. Februar 2001
Tag der Abgabe : 26. Juli 2001

Referent	:	Prof. Dr. Günter Last
Korreferent	:	Priv. Doz. Dr. Dieter Kadelka
Betreuer	:	Dr. Michael Beigl

In Kooperation mit
Fakultät für Informatik, Institut für Telematik, Telecooperation Office

Ich erkläre hiermit, daß ich die vorliegende Arbeit selbständig verfaßt und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe.

Cand.-Math. techn. Andreas Neumann MSc
E-Mail: andreas@aneumann.de
WWW: http://www.aneumann.de/

Karlsruhe, den 26. Juli 2001

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Mathematische Notation

$\mathbb{N} = \{1, 2, \ldots\}$ = die Menge der natürlichen Zahlen
$\mathbb{R} = (-\infty, +\infty)$ = die Menge der reellen Zahlen
$\mathbb{R}^+ = [0, \infty)$
$\mathbb{Z} = \{\ldots, -1, 0, 1, \ldots\}$ = die Menge der ganzen Zahlen
$\mathbb{Z}^+ = \{0, 1, \ldots\}$
$\bar{\mathbb{Z}}^+ = \{0, 1, \ldots\} \cup \{+\infty\}$
$|\mathbb{X}|$ = die Kardinalität der Menge $\mathbb{X}$
$\mathcal{B}^+$ = die Borelsche σ-Algebra in $\mathbb{R}^+$
$\mathbf{1}_{\{\cdot\}}$ = Indikatorfunktion von $\{\cdot\}$

1. Einleitung

1.1 Ubiquitous Computing

In unserer immer mehr von Computern abhängigen Welt wird es gleichzeitig immer schwieriger, mit ihnen umgehen zu können. Der Lernaufwand steigt jenseits des für viele Menschen zeitlich machbaren und für ihre Arbeit sinnvollen Rahmens. Dieses hängt nicht mit der immer höheren technischen Komplexität der Systeme zusammen, sondern liegt im falschen Ansatz bei der Entwicklung der Benutzerschnittstellen begründet. Anstelle daß sich der Mensch in seinen Tätigkeiten der Maschine anpaßt, wie es bei dem heute am verbreitetesten Computertyp, dem Personal Computer, der Fall ist, sollte er Maschinen bauen, die besser seinen Verhaltensweisen und im Alltagsleben trainierten Interaktionsformen entsprechen. Der Computer von heute ist trotz seines Netzwerkanschlusses vom realen Raum um ihn herum weitestgehend isoliert und nicht in der Lage, sich auf verschiedene Situationen einzustellen. Daher befindet er sich zu häufig im Fokus unserer Aufmerksamkeit anstelle als Dienstleister im Hintergrund unsere Tätigkeiten zu unterstützen.

Um Interaktionsstrukturen zu schaffen, bei denen nicht der Computer sondern der Mensch im Zentrum steht, bedarf es einer anderen Art von Computertechnologie. Diese wird von Mark Weiser als die Technologie der dritten Phase bezeichnet. Tabelle 1.1 gibt einen Überblick über die Einteilung der Computerentwicklung, wie er sie auf seiner Homepage [21] darstellt. Die dritte Phase ist die des Ubiquitous Computing (ubiquitous = allgegenwärtig). Weiser prägte diesen Begriff 1988 bei seiner Arbeit am Xerox Palo Alto Research Center und beschrieb ihn in seinem wegweisenden Artikel „Der Computer des 21. Jahrhunderts“ [22]. Diese Phase zeichnet sich durch viele Computer pro Person aus, deren Stärken darin liegen, dem Menschen allgegenwärtig zu dienen, ohne unnötig Aufmerksamkeit auf sich zu lenken, geschweige denn, ihn zu zwingen, sich ihnen in seinen Tätigkeiten anzupassen. Das wird erreicht durch drahtlose Vernetzung vielleicht irgendwann hunderter stark spezialisierten Computer pro Person. Jeder einzelne dieser Rechner ist verglichen mit einem heutigen Desktop Computer sehr beschränkt in seinen Fähigkeiten, häufig sogar einer bestimmten Tätigkeit zugeordnet. Mit Hilfe der drahtlosen Kommunikation sind sie jedoch in der Lage, dem Menschen im Zusammenspiel wesentlich besser zur Seite zu stehen.

Phase	Technologie	Charakteristika
I	Großrechner	ein Rechner für viele Personen
II	Personal Computer	ein Rechner pro Person
III	Ubiquitous Computing	viele Rechner pro Person

Tabelle 1.1: Trends der Computertechnologie nach Weiser [21]

Wir befinden uns zur Zeit in der Anfangsphase dieser dritten Generation von Rechnern, wie z. B. durch die vermehrte Verbreitung von Laptops, Persönlichen Digitalen Assistenten (PDA) und Mobiltelefonen, die häufig mehr Funktionen als das bloße Telefonieren anbieten, zu erkennen ist. Bei diesen Geräten lassen sich bereits viele Trends zum Ubiquitous Computing beobachten, als da wären, die Mobilität und damit der Einsatz in vielen Alltagssituation weit weg vom Schreibtisch, die höhere Spezialisierung zu bestimmten Tätigkeiten, die drahtlose Kommunikation durch Infrarot oder Funk und die Vernetzung der Geräte durch WAP, Bluetooth, etc. Die Umwelterfassung ist bei diesen Geräten noch nicht sehr weit fortgeschritten, je mehr sie jedoch auf bestimmte Funktionen ausgerichtet sind, desto besser kann man das Gerät auf die typischen Situationen, in denen es benutzt wird, abstimmen. Diese spezialisierten Geräte werden unter dem Begriff der Information Appliances[1] zusammengefaßt.

1.2 Motivation

Einen großen Teil unserer Arbeitszeit verbringen wir mit dem Sichten von Informationen. Informationen werden, um sie zeitlich oder örtlich zu transportieren, in Dokumenten gespeichert. Ein Dokument setzt sich somit aus seinem Inhalt, der die Information übermittelt, und dem Transportmedium zusammen. Transportmedien, die der Informationsübermittlung durch die Zeit dienen, werden auch Speichermedien genannt. Herkömmlich wurden z. B. Text- und Bildinhalte vorwiegend auf Papier und bewegte Bilder mit Hilfe von Filmrollen transportiert. Durch die Verbreitung des Internets liegen uns vermehrt Inhalte in digitalen Medien als Texte, Bilder, Videos oder Audios vor. Diese Medien haben jedoch den Nachteil, daß sie im Gegensatz zu auf Papier gedrucktem Text ihren Darstellungsmechanismus nicht selber mitbringen, sondern für jedes Medium eine spezielle Abspielmöglichkeit, d. h. ein Ausgabegerät, welches den Medientyp darstellen kann, gesucht werden muß.

In vielen Fällen ist die Verweildauer bei einem Dokument, d.h. die Zeit, die wir uns z. B. mit einem Text beschäftigen, gering, da es sich dabei häufig um kurze Dokumente handelt oder wir uns nur einen schnellen Überblick verschaffen wollen. Daher trägt die Zeit, die wir benötigen, eine geeignete Abspielmöglichkeit in unserer momentanen Umgebung zu suchen und das entsprechende Dokument zu transferieren, einen erheblichen Anteil an der Gesamtzeit, die wir mit einem Dokument verbringen.

[1] Eine genaue Definition und weitere Beispiele sind zu finden in [13].

Diese Zeit jedoch ist verloren für unsere eigentliche Aufgabe, die wir mit Hilfe der Information des Dokumentes bearbeiten. Weiterhin werden wir unnötigerweise vom Inhalt unserer Arbeit durch die Beschäftigung mit technischen Problemen, wie z. B. der Bedienung eines DVD-Players oder der eines speziellen MP3-Spielers, abgelenkt. Diese Problematik tritt noch verstärkt in uns unbekannten technischen Umgebungen zu Tage, z. B. im Büro einer Kollegin oder beim Besuch eines Freundes. Es ist eine Aufgabe der Technik, uns überall den Zugang zu Informationen so leicht wie möglich zu gestalten.

Bisher jedoch ist mit jedem neuen Datenformat, unterschiedlichen Benutzerschnittstellen und immer komplizierteren Geräten eher eine Verschlechterung eingetreten. Wenn wir uns ein Dokument, z. B. einen Film aus dem Internet, anschauen möchten, sollte uns die Technik nicht nur das Finden eines geeigneten Abspielgerätes in unsere Nähe abnehmen, sondern dieses auch automatisch mit der Darstellung beauftragen. Besonders in Hinblick auf die in Abschnitt 1.1 angesprochene dritte Rechnergeneration, die zur Zeit Einzug in unsere Alltagswelt hält, wird die Benutzung eines solchen Automatismus unabdingbar. Mit der zunehmenden Anzahl von vernetzten potentiellen Ausgabegeräten pro Mensch wächst die Komplexität und damit der Aufwand, diese sinnvoll einzusetzen. Ein Mensch fühlte sich nicht nur schnell überfordert, sondern der von ihm nun zu errichtende Verwaltungsaufwand führe die gesamte Idee der Effizienzsteigerung ad absurdum. Die Konsequenz könnte sein, daß er sich auf wenige Ausgabegeräte beschränken und nicht die volle Kapazität seiner lokalen Umgebung nutzen würde. Dieses ist der wichtigste Aspekt der Motivation, aufgrund dessen diese Arbeit im Bereich Ubiquitous Computing einzuordnen ist. Durch den Einsatz eines automatischen Verteilungssystems wird aus den den Menschen umgebenden Einzelgeräten eine zusammenhängende Ausgabeumgebung, die ihm dient, ohne daß er sich ihres technischen Aufbaus bewußt sein muß.

Die entscheidende Fähigkeit eines solchen Systems ist die, die Medienströme richtig auf die Ausgabegeräte zu verteilen. Geräte nach ihren technischen Möglichkeiten zur Ausgabe bestimmter Medientypen grob zu klassifizieren ist relativ einfach. Unter Zuhilfenahme von Kontextinformationen über die Situation des Benutzer muß jedoch durch Auswahlmechanismen jenes Gerät gefunden werden, welches der Benutzer selber als das geeigneteste anerkennt. Mißlingt dies, fühlt sich der Mensch bevormundet und die Akzeptanz des Systems sinkt. Die Komplexität des Verteilungsproblems kann man sich vor Augen führen, wenn man bedenkt, daß sich nicht nur ein Mensch häufig mehrerer Medien gleichzeitig bedient, sondern, wie z. B. in einem Büro, mehreren Menschen viele Ausgabegeräte gemeinsam zur Verfügung stehen, so daß es zu einem Wettstreit um diese Ressource kommt. Um das System auf diese Anforderungen abzustimmen, ist eine mathematische Untersuchung derartiger Konfliktfälle nötig. Diese Untersuchung bildet neben dem Entwurf einen Schwerpunkt der Arbeit, da sie die Grundvoraussetzung für das Funktionieren eines solchen Systems darstellt. Hierbei wird die exakte mathematische Folgerung, die nur in einfachen Fällen möglich ist, in komplexeren Fällen durch eine Simulation anhand eines stochastischen Modells ersetzt. Nur dadurch ist es möglich, die Auswahlmechanismen so abzustimmen, daß das System zur Zufriedenheit der Benutzer arbeitet und jedem die volle Rechen- und Ausgabekapazität der Umgebung zur Verfügung stellt.

Das in dieser Arbeit beschriebene System ist nach dem griechischen Götterboten und Nachrichtenübermittler Hermes auf den Namen HermiUC (Hermes in Ubiquitous Computing) getauft. Wie HermiUC das Arbeiten erleichtern kann, soll an folgendem Anwendungsszenarium aufgezeigt werden.

1.3 Anwendungsszenarium

Es hat sich häufig als schwierig erwiesen vorauszusagen, wie der Mensch ein neues Hilfsmittel tatsächlich einsetzt. Dennoch soll hier der Versuch gewagt werden, um die Möglichkeiten, die dem System innewohnen, anschaulich darzustellen.

Herr Meier kommt in sein Büro. Gestern abend hat er einen Bericht im Fernsehen gesehen, den er seiner Kollegin Frau Huber zeigen möchte. Er geht in ihr Büro, nimmt seinen PDA aus der Hemdtasche und tippt auf die Internetadresse, die er sich gestern während der Sendung notiert hat. Gemeinsam schauen sie auf die elektronische Tafel, auf der in diesem Moment automatisch der Bericht, der aus dem Internet heruntergeladen wird, zu laufen beginnt. HermiUC hatte ihn in ihrem Büro lokalisiert, und ein von ihr gerade nicht benötigtes geeignetes Ausgabegerät, die Tafel, mit der Darstellung seiner Anfrage beauftragt.

Anschließend beim Kaffeeholen trifft er Herrn Kiefer. Dieser möchte mit ihm den aktuellen Stand eines gemeinsamen Projektes durchgehen. Sie gehen in Meiers Büro. Herr Kiefer wählt auf seinem PDA das Projektdokument aus, welches sogleich auf Meiers Computerbildschirm erscheint. Während sie es gemeinsam durchgehen, fällt Herrn Kiefer eine Skizze ein, die man verwenden könnte. Er wählt sie auf seinem Computer aus. Sie wird augenblicklich auf dem Bildschirm, der vor seinem Schreibtisch in die Wand eingearbeitet ist, dargestellt. So können sie gleichzeitig das Dokument bearbeiten und die Skizze betrachten, ohne daß sich diese auf dem Computerbildschirm gegenseitig überdecken. Das System hatte die Belegung des Monitors erkannt, stattdessen ein naheliegendes freies Ausgabegerät gewählt und das von beiden offensichtlich erwartete getroffen.

Es klopft und freudestrahlend kommt Frau Radke herein. Sie tippt auf ihrem Mobiltelefon und schon ertönt aus den Lautsprechern in Meiers Büro ein Gesprächsmitschnitt von ihr mit einem Kunden. Die Firma hat einen neuen Auftrag! Da drei Personen sich in einem relativ kleinen Raum befinden, der keinen Platz zur individuellen Arbeit aller bietet, ging das System von einer Versammlung aus und machte Frau Radkes Anfrage für alle Anwesenden hörbar.

Es wird eine Besprechung im Sitzungszimmer anberaumt. Die Teilnehmer sollen bis dahin jeder für sich eine Liste von Vorschlägen den Auftrag betreffend erstellen. Jeder bringt seine bevorzugte Benutzerschnittstelle mit. Frau Radke hat ihr Mobiltelefon mit Internetzugang, Herr Meier und Frau Huber ihre PDAs und Herr Kiefer seinen Laptop. Herr Kiefer, der die Sitzung leitet, wählt auf seinem Laptop die Informationen mit den Eckdaten zum Vertrag aus, die sofort auf einem der sich im Raum befindlichen Großbildschirme, für alle einsehbar, dargestellt werden. Frau Huber wählt ihre Datei mit der Vorschlagsliste aus, die auf einer anderen elektronischen Tafel erscheint. Man beschließt, eine gemeinsame Liste der brauchbaren Vorschläge zu erstellen, die Herr Kiefer in seinem Laptop bearbeitet und für alle ersichtlich im Raum anzeigen läßt. Herr Meier, der seine Ideen als nächster ansprechen möchte, beauftragt HermiUC mit der Darstellung. Das System, das eine Besprechungssituation erkennt, legt seinen Wunsch in eine Warteschlange, da kein für alle einsehbares Ausgabegerät im Raum mehr frei ist. Nachdem man Hubers Liste diskutiert hat, tippt sie erneut auf ihren PDA, ihre Liste erlischt und Meiers wird auf der nun freigewordenen Tafel angezeigt.

Betont werden sollte, daß es sich bei diesem Büroszenarium nur um eine Anwendungsmöglichkeit handelt. Mit der Zunahme von intelligenten Alltagsgegenständen und deren Vernetzung hält das Ubiquitous Computing auch im Haushalt Einzug. Dort kann dieses System in naher Zukunft z. B. in der Küche Konflikte bei der Darstellung der elektronischen Zeitung für den Vater, den Fernsehnachrichten für die Tochter und den morgendlichen Staumeldungen aus dem Internet für die Mutter lösen.

1.4 Aufbau der Arbeit

Zunächst werden in Kapitel 2 die Grundlagen zur Entwicklung des Systems erarbeitet. Es werden Fehlentwicklungen der Computertechnik im Hinblick auf die Abstimmung auf den Menschen aufgezeigt. Danach wird die derzeitige Situation des Umgangs mit Medien und Dokumenten verschiedenen Typs untersucht, um daraus Implikationen auf den Entwurf des Systems zu schließen. Die Analyse ist bewußt knapp gehalten. Die wichtigsten Einflußfaktoren und Konzepte werden dargestellt, um den Gesamtzusammenhang verständlich zu machen, in dem der Schwerpunkt der Arbeit steht.

Kapitel 3 beschäftigt sich mit dem theoretischen Aufbau des Systems. Der Entwurf wird anhand der Vorgaben aus Kapitel 2 vorgenommen. Zunächst werden die benötigten Komponenten der Infrastruktur besprochen: die Eingabegeräte, der Controller und die Ausgabegeräte. Dann wird auf die Funktionalität des Gesamtsystems eingegangen. Diese umfaßt die Verarbeitung der Lokationsinformationen, den Auswahlmechanismus zur Bestimmung der Ausgabegeräte sowie die eigentliche Medienausgabe. Das Kapitel schließt mit der Darstellung der Kommunikationsabläufe zwischen den Komponenten, die zum Betrieb notwendig sind.

In Kapitel 4 wird ein stochastisches Modell basierend auf dem Aufbau aus Kapitel 3 zur mathematischen Beschreibung des Systems erstellt. Mit Hilfe dieses Modells wird die Verteilung der Ausgabeströme mathematisch untersucht und im Folgenden simuliert, um sie zu optimieren und Voraussagen über das Verhalten beim Betrieb treffen zu können. Zunächst werden einfache Spezialfälle mathematisch analysiert und, wie in Kapitel 5 beschrieben, im Computer simuliert, um durch Übereinstimmung der Ergebnisse die Korrektheit der Simulationstechnik abzusichern. Danach wird die Simulation der Büroumgebung des Telecooperation Office (TecO) der Universität Karlsruhe als realistisches Anwendungsszenarium beschrieben und im Folgenden die sich daraus ergebenden Rückschlüsse auf den Entwurf.

In Kapitel 6 wird eine mögliche Implementierung und Eingliederung des HermiUC-Systems in existierende Anwendungen am TecO vorgestellt, um die Realisierbarkeit nachzuweisen. Die Umsetzung einer Implementierung ist aufgrund des Umfangs nicht Teil dieser Arbeit.

Zuletzt werden in Kapitel 7 Ausblicke auf mögliche weiterführende Untersuchungen und Erweiterungen des Systems gegeben. Ebenso ist hier eine Zusammenfassung und Danksagung enthalten.

2. Situationsanalyse

2.1 Stand der Computertechnik

Dem PC Paradigma entsprechend wurde er nicht nur zum Standardausgabegerät für alle Medientypen ausgebaut, auch die gesamte Ausgabenkontrolle läuft über ihn ab. Dieses führt nicht nur zum Problem der Bedienungskomplexität, sondern es handelt sich dabei auch um ein äußerst starres System. Generell erhöht die Trennung von Kontrolle und Ausgabe die Flexibilät von Systemen und stellt somit mehr Wege zur Verfügung, sie an die Bedürfnisse der Benutzer anzupassen [6]. Weiterhin haben sich in der Vergangenheit Infrastrukturen in der Computertechnologie als langlebiger und deutlich weniger wechselhaft als spezielle Gerätetechnologien erwiesen. Es sollte somit so weit wie möglich auf Web Komponenten bei der Entwicklung zurückgegriffen werden, da diese Technologien eine längere Einsatzfähigkeit des Systems auch aufgrund ihrer Unterstützung auf immer neuen Geräten gewährleistet.

2.1.1 Benutzerschnittstellen

Zunächst muß zwischen zwei verschiedenen Interaktionsformen getrennt werden. Bei der expliziten Interaktion wählt der Benutzer direkt eine Multimedia-Datei aus, um sie sich anzeigen zu lassen. Implizite Interaktionen liegen dann vor, wenn Anfragen an das System automatisch aus anderen Anwendungen heraus generiert werden, z. B. beim Auswählen eines Links in einem Web-Browser [6]. Hierbei liegt als Schnittstelle bereits eine andere Anwendung vor, die auf das hier besprochene System als Dienstnehmer zugreift. Eine eigene Benutzerschnittstelle liegt daher nur der expliziten Interaktion zugrunde, weshalb in dieser Arbeit benutzerschnittstellenspezifische Ausführungen immer von diesem Fall ausgehen. Es soll jedoch betont werden, daß die automatische Verteilung auch als Dienst anderen Anwendungen bereit gestellt werden kann.

Die Anzahl der zur Verfügung stehenden potentiellen Ein- und Ausgabegeräte hat sich in den letzten Jahren deutlich gesteigert. Dieses läßt sich zum einen auf die Erhöhung der Rechengeschwindigkeit zurückführen, die neue Arten der Eingabe, z. B. natürliche Sprache, ermöglicht. Zum anderen stehen aufgrund ihrer drahtlosen Kommunikationsfähigkeit auch viele neue Geräte der dritten Rechnergeneration als

Ein- und Ausgabegeräte zur Verfügung (vgl. Abschnitt 1.1). Hiervon wurden im Anwendungsszenarium (Abschnitt 1.3) bereits Mobiltelefone und PDAs erwähnt. Der Vorteil dieser persönlichen Geräte ist, daß der Benutzer jederzeit mit einer ihm bekannten Schnittstelle arbeitet. Weiterhin erlauben sie die lokale Speicherung von sensitiven Daten, was den Schutz der Privatsphäre erleichtert. Darauf wird noch weiter im Abschnitt 2.1.2 eingegangen.

Eine Zielsetzung des hier vorgestellten Systems ist es, den Umgang mit technischen Geräten im Gegensatz zu heute deutlich zu vereinfachen und wie Donald A. Norman es in seinem Buch „The invisible Computer“ [13] ausdrückt, man sollte in der Lage sein, das Werkzeug zur Arbeit mitzunehmen, und nicht wie heute die Arbeit zum Werkzeug bringen zu müssen. Dieses läßt sich mit mobilen Schnittstellen konsequent realisieren. Zu beachten ist, daß mit jeder zusätzlichen Funktion, die dem Benutzer angeboten wird, die Komplexität in der Bedienung steigt. Im Jahre 1992 hatte das Textverarbeitungsprogramm Microsoft Word 311 Befehle, bereits fünf Jahre später war diese Zahl auf 1.033 angestiegen [13]. Hier liegt eindeutig ein falsches Konzept zugrunde. Es werden alle technisch realisierbaren Funktionen eingebaut, solange sie auch nur eine einzige Person für sinnvoll hält. Diese Überfunktionalität, die zur Überforderung des Benutzers führt, hat auch wirtschaftliche Gründe, die Donald A. Norman in dem Kapitel „What´s Wrong with the PC?“ [13] darstellt. Auf diese Gründe wird an dieser Stelle nicht weiter eingegangen. Bei dem hier besprochenen System geht es darum, einfache Bedienung zu bieten, so daß es ohne Erklärung sinnvoll eingesetzt werden kann. Hierbei ist eine Reduzierung auf die notwendigsten Funktionen unbedingt gewollt.

Als Schnittstelle bietet sich daher ein Waba-Programm an, welches auf den persönlichen PDAs und Mobiltelefonen den Zugang zu allen in der lokalen Umgebung zur Verfügung stehenden Ausgabegeräten ermöglicht. Diese Eingabegeräte können damit ebenfalls die automatische Lokalisation ihres Benutzers unterstützen [1].

Implikationen auf den Entwurf:

Für die explizite Interaktion mittels persönlicher mobiler Geräte sollte ein universelles, geräteunabhängiges Benutzerprogramm entworfen werden. Dieses stellt einen deutlichen Fortschritt gegenüber heutigen Systemen dar, wo sich der Mensch bei jedem Gerät auf eine neue Oberfläche einstellen muß. Der Designschwerpunkt muß in der einfachen Bedienung liegen. Es muß jederzeit vorsichtig abgewägt werden, ob eine erweiterte Funktionalität die damit verbundene Erhöhung der Komplexität der Bedienung gerechtfertigt.

2.1.2 Sicherheitsaspekte

Es ist wichtig zu beachten, daß es keine rein technische Lösung zur Sicherung der Privatsphäre geben kann [23]. Soziale Aspekte z. B. im Berufsumfeld spielen eine wichtige Rolle. Zur sicheren Kommunikation zwischen den Geräten stehen heutzutage ausreichende kryptographische Verfahren zur Verfügung. Die kritischen Punkte dieses Systems sind zum einem die notwendige Positionsbestimmung der Benutzer und zum anderen die Überwachung der Anfragen, die ein Benutzer stellt. Hier muß besonders darauf geachtet werden, daß nicht mehr als nötig gespeichert wird und die Daten ausschliesslich autorisierten Instanzen zur Verfügung gestellt werden, um sich so weit wie möglich vom Schreckensszenarium eines Überwachungssystems zu distanzieren. Interessant ist hier das aus dem Internet bekannte Konzept der digitalen Pseudonyme. Man muß wissen, welche Anfrage jemand gestellt hat und wo sich dieser jemand in dem Moment befindet, in dem die Darstellung beginnt, aber wer dieser jemand wirklich ist, spielt für die Erfüllung der Anfrage keine Rolle. Als problematisch erweist sich dieser Ansatz, wenn es sich um Zugriffe auf gesicherte Daten handelt, für die eine Authentifizierung notwendig ist. Da es sich hierbei jedoch um kein spezifisches Problem handelt, welches der automatischen Medienverteilung entspringt, wird beim Enwurf nicht weiter darauf eingangen. Auf Erweiterungen des Systems in dieser Hinsicht wird in Abschnitt 7.2 eingegangen. Ein Interessenkonflikt kann zwischen der Sicherung der Privatsphäre und dem Speichern von Daten dann auftreten, wenn man zur Verbesserung des individuellen Service durch das System Benutzerprofile einsetzt. Das Design bezüglich der Sicherung der Privatsphäre spielt eine große Rolle bei der Akzeptanz des Systems durch seine Benutzer.

Implikationen auf den Entwurf:

Es sollten bei der Entwicklung Möglichkeiten zum Schutz der Privatsphäre berücksichtigt werden. Wie diese jedoch angewandt werden, hängt zum großen Teil von der sozialen Einstellung der Benutzer ab. Generell sollten jedem Menschen alle vertraulichen Daten über ihn selbst gehören, unabhängig davon, wer die Maschine besitzt auf der sie gespeichert sind. Der Entwurf sollte erweiterbar im Hinblick auf Authentifizierungsmechanismen sein. Es sollte soweit wie möglich mit digitalen Pseudonymen gearbeitet werden. Auf die Erstellung von Benutzerprofilen wird verzichtet, da der Privatsphäre der eindeutige Vorrang gewährt wird.

2.2 Medienverteilung

Der Umgang mit Dokumenten und damit die richtige Wahl eines Ausgabegerätes hängt im großen Maße vom Medientyp ab. Tabelle 2.1 zeigt eine Klassifizierung der Medientypen im Hinblick auf die Kommunikationsart, Zeitabhängigkeit und Darstellungslänge. Diese Klassifizierung wird im Verlauf der Arbeit dem Entwurf zugrunde gelegt.

Definition 2.1. Eine für den Benutzer dargestellte Information, die nicht der Steuerung des Systems dient, wird als *Ausgabestrom* bezeichnet.

Art	Zeitabhängigkeit	Länge	Typnummer	Beispiel
Audio	nichtstatisch	unbegrenzt	1	Radio
		begrenzt	2	Tondokument
Audiovisuell	nichtstatisch	unbegrenzt	3	Fernsehen
		begrenzt	4	Tonfilm
Visuell	statisch	unbegrenzt	5	Bild/Text
	nichtstatisch	unbegrenzt	6	Webcam
		begrenzt	7	Stummfilm

Tabelle 2.1: Klassifizierung der Medientypen

Ausgabestrom dient somit als Oberbegriff der Darstellung aller Medienarten, unabhängig davon, ob es sich um eine kontinuierliche Ausgabe handelt, wie es bei Radio- oder Fernsehprogrammen der Fall ist, oder um eine einmalige Informationsübermittlung wie bei statischen Texten oder Bildern.

2.2.1 Verwandte Arbeiten

Es gibt eine Vielzahl an Projekten, die sich mit Teilbereichen des hier besprochenen Systems beschäftigen. Einige werden hier kurz dargestellt und gegenüber der vorliegenden Arbeit abgegrenzt. Betont werden sollte, daß zum Schwerpunkt dieser Arbeit, der Verteilung der Ausgabeströme, bisher kaum Veröffentlichungen geschehen sind. Das Konzept wurde im Rahmen des UbicompBrowser-Projektes am Telecooperation Office der Universität Karlsruhe angesprochen [6]. Jedoch wurde der dort propagierte einfache regelbasierte Ansatz nicht zu einem vollständigen Entwurf weiterentwickelt, so daß sich keine qualitativen Aussagen machen lassen. Aspekte der vorliegenden Arbeit, die bereits an anderer Stelle behandelt worden sind, beinhalten:

1. Lokalisierungsmechanismen
2. Integrierte Computerumgebungen mit verschiedenen Ein- und Ausgabegeräten
3. Situationserfassung mit Hilfe von Kontextinformationen
4. Architekturen für Ubiquitous Computing Umgebungen

Zum 1. Punkt, einem kritischen Bestandteil jeder Ubiquitous Computing Umgebung, gibt es unzählige Untersuchungen, Prototypen und sogar bereits kommerzielle Produkte, wie z. B. die Active Badges von Olivetti Research [9]. Es ist für den Entwurf unerheblich, welche der zur Verfügung stehenden Techniken bei der Implementierung eingesetzt wird, solange ein Mindestmaß an Positionsgenauigkeit, d.h. eine vorgegebene Feinheit der Granularität, eingehalten wird. Wichtig bleibt festzuhalten, daß es zwar Studien mit Videokameras zur Bestimmung der Ausrichtung, sprich der

Bickrichtung einer Person gibt, diese jedoch aufgrund der häufigen Blickrichtungsänderungen beim Menschen nur unzuverlässig in das hier besprochene System zu integrieren wären. Daher wird darauf verzichtet.

Eines der am weitest fortgeschrittenen Projekte in Hinblick auf den 2. Punkt ist die Interactive Room (iRoom) Infrastruktur, die an der Stanford University entwickelt wird [8]. Dort werden verschiedene Ein- und Ausgabegeräten von PDAs bis zu elektronischen Wandtafeln zu einer zusammenhängenden Arbeitsumgebung, dort speziell ein Besprechungszimmer, integriert. Die Verteilung der Medien auf die Ausgaberäte erfolgt dort manuell über einen sogenannten „room display manager". Mit dessen Hilfe kann man z. B. per drag and drop von URLs auf einer schematischen Darstellung der Ausgabegeräte des Raumes das Anzeigen auslösen. Ein Beispiel aus dem Privatbereich ist der AwareHome [11] Prototyp am Georgia Institute of Technology. Dabei handelt es sich um ein ganzes Haus, welches in Möbeln und Wänden mit Sensoren und Computer ausgestattet ist und als bewohnbares Laboratorium fungiert.

Heutige Desktopcomputer sind nicht einmal in der Lage festzustellen, ob der Benutzer noch vor dem Bildschirm sitzt oder schon Mittagessen gegangen ist. Die Verwendung von Kontextinformationen zur Situationserfassung des Benutzer ist der entscheidende Punkt, wenn es darum geht, den Menschen durch Dienste zu unterstützen, ohne seine Aufmerksamkeit an sich zu reißen. Der 3. Punkt, die Verwertung von Kontextinformationen, die über die Positionserfassung hinausgeht, befindet sich heutzutage noch in einem Anfangsstadium, birgt jedoch ein großes Potential, wie es in dem Artikel „There is more to Context than Location" [16] von Albrecht Schmidt u. a. beschrieben ist. Dieses gilt auch für das hier vorgestellte System. Die Qualität des Dienstes würde enorm erhöht, wenn z. B., wie im Anwendungsszenarium (Abschnitt 1.3) beispielhaft dargestellt, Besprechungen automatisch erkannt würden und alle Ausgaben auf für jeden einsehbare Großbildschirme umgeleitet würden. Um dieses zu leisten und gleichzeitig für Ausbaumöglichkeiten offen zu bleiben, sollte die Kontextauswertung in einem separaten Modul erfolgen, auf das der Auswahlmechanismus zur Bestimmung der Ausgabegeräte zugreifen kann.

Der letzten Punkt ist eng mit dem 2. verknüpft. Neben dem schon dort angesprochenen iRoom-Projekt, sollte auf das am TecO entwickelte RAUM-System hingewiesen werden [10]. Dabei handelt es sich um eine Architektur, die speziell auf ortsabhängige lokale Kommunikation, wie sie in Ubiquitous Computing Umgebungen auftritt, ausgerichtet ist.

Implikationen auf den Entwurf:

Es sollten Intergrationsmöglichkeiten für ein spezielles erweiterbares Kontextmodul offen gehalten werden, auf das das System über definierte Schnittstellen zugreifen kann. Es bietet sich an, das System auf der RAUM-Architektur aufzubauen, um von der schon bestehenden Struktur zu profitieren und die automatische Verteilung der Ausgabeströme anderen Anwendungen zur Verfügung zu stellen.

2.2.2 Mediendarstellung

In einigen Fällen wie z. B. bei Sportübertragungen oder Diskussionsrunden könnte man erwägen, Tonfilme ohne Ton oder nur den Ton anzubieten, wenn ansonsten aufgrund eines hohen Belegungsgrades lange Wartezeiten bis zur Darstellung in Kauf genommen werden müssten. Dabei handelt es sich jedoch um keine befriedigende Dienstleistung für den Benutzer, da im allgemeinen davon auszugehen ist, daß er den Film nicht schon bereits kennt und so den Einfluß der einzelnen Medienkomponenten auf das Gesamtverständnis einschätzen kann. Dagegen sind, was die Darstellungsqualität betrifft, Kompromisse durchaus gewünscht, um Wartezeit und räumliche Distanz zum Ausgabegerät zu minimieren. Dieses tritt besonders im Bereich visueller Medien auf, die mit Abstrichen in der Qualität jedoch ohne Informationsverlust auf Bildschirmen verschiedener Größe und Auflösung dargestellt werden können. Dabei sollte jedoch eine untere Schranke beachtet werden, die die schlechteste noch zu akzeptierende Qualität darstellt. Neben der Darstellungsqualität ist auch die Darstellungsdauer zu berücksichtigen. Neben zeitlich befristeten Medien wie Filmen und Audioaufnahmen gibt es unbefristete Medien wie Bilder, Texte und Liveübertragungen. Bei ersteren wird die Darstellung mit dem Ende gestoppt, bei den anderen wäre sowohl eine automatische Ermittlung des Darstellungsende als auch eine manuelle Beendigung durch den Benutzer möglich. Da die Kontexterfassung noch nicht so weit fortgeschritten ist, daß das System mit genügend hoher Sicherheit feststellen kann, welche Darstellungen noch benötigt werden, wird von einer automatischen Beendigung abgesehen. Die Frustrationsgrenze beim Benutzer würde beim wiederholten Stoppen noch benötigter Darstellungen schnell überschritten.

Ein besonderes Problem tritt bei Audioströmen auf. Im Gegensatz zu visuellen Darstellungen läßt sich Audio nur bedingt gerichtet ausgeben, so daß es bei gleichzeitiger Ausgabe mehrerer Audiodokumente innerhalb eines Raumes zu Störungen kommen kann. Zur Erkennung derartiger Situationen bedarf es eines sehr ausgereiften Kontextmoduls, da die Lautstärke, die Ausgaberichtung, die Personen im Raum, Gespräche, etc. eine Rolle spielen können. Solange die Interpretation derartiger Szenarien nicht mit ausreichender Verläßlichkeit möglich ist, sollten der gleichzeitigen Audioausgabe keine speziellen Restriktionen auferlegt werden, da im Zweifelsfall von der Verwendung von Kopfhörern ausgangen werden kann.

Implikationen auf den Entwurf:

Die Auslassung von Medienkomponenten bei kombinierten Medien wie z. B. Tonfilmen wird nicht berücksichtigt. Jedem Ausgabegerät sollte eine Liste von Medientypen zugewiesen werden, die es in akzeptabler Qualität darstellen kann. Das Beenden der Darstellung von Medientypen mit unbegrenzter Länge geschieht ausnahmlos auf Befehl des Anfragestellers. Alle Medientypen einschließlich Audio werden bei der Auswahl der Ausgabegeräte gleich behandelt.

2.2.3 Anfrageabhängigkeiten

Es liegt nahe, Beziehungen zwischen Anfragen bzgl. ihres Medientyps zu untersuchen. So ist es heutzutage häufig der Fall, daß Videoanfragen aus einem Textdokument hervorgehen, das auf den entsprechenden Film verweist. Hier wird jedoch die These vertreten, daß Abhängigkeiten dieser Art den heutigen technischen Umgebungen entspringen, die zur Medienwiedergabe genutzt werden und keine den Medientypen inhärente Eigenschaften sind. Diese Abhängigkeiten werden in einer Ubiquitous Computing Umgebung verschwinden. Wenn sich z. B. durch Barkodeleser in PDAs Adressen von Informationsfilmen oder gesprochenen Bedienungsanleitungen direkt von Konsumprodukten lesen lassen, wird die Vorherrschaft der Textmedien zur Übersichtsvermittlung weiterer Informationen verschwinden. Der Vollständigkeit halber wird erwähnt, daß es zeitliche Abhängigkeiten sehr wohl gibt. Eine Vielzahl von Anfragen in kurzer Zeit läßt auf eine derzeit hohe Aktivität im Büro schließen, aus der sich eine erhöhte Wahrscheinlichkeit weiterhin vieler Anfragen ergibt.

Implikationen auf den Entwurf:

Die stochastisch vorgenommene Bestimmung der Medientypen in der Simulation erfolgt unabhängig von der Vergangenheit und vom Zeitpunkt der korrespondierenden Anfrage. Es wird somit jeder erzeugten Anfrage mit Hilfe mathematischer Verteilungen von Grund auf neu ein Medientyp zugordnet, der weder vom Anfragezeitpunkt noch von den Zuordnungen zu vorherigen Anfragen abhängig ist.

3. Gesamtaufbau des Systems

In diesem Kapitel wird der theoretische Aufbau des Systems beschrieben. Er richtet sich nach den Vorgaben des vorangehenden Kapitels. Weiterhin liegt dem Entwurf die Restriktion zugrunde, daß das entwickelte System in seinen wichtigsten Punkten, speziell der Verteilung der Ausgabeströme, stochastisch modellierbar bleiben muß. Diese Bedingung schränkt den Aufbau nicht ein, sondern beeinflußt lediglich die Entscheidung bei alternativen Möglichkeiten, z. B. zwischen der Verwendung von mobilen und nichtmobilen Ausgabegeräten. Alle diese Entscheidungen lassen sich jedoch ebenso aus der Designperspektive begründen. Es wird daher nicht immer speziell darauf hingewiesen. Alternative Konzepte und Ausbaumöglichkeiten werden später in Kapitel 7 vorgestellt.

3.1 Gerätekomponenten

3.1.1 Eingabegeräte

Nach Abschnitt 2.1.1 sollte eine universelle, geräteunabhängige Benutzerschnittstelle verwendet werden. Sie sollte auf möglichst vielen mobilen persönlichen digitalen Geräten zur Verfügung stehen. Wie im Anwendungsszenarium (Abschnitt 1.3) beschrieben, läßt sich das System dann z. B. durch PDAs oder Mobiltelephone steuern. Gemäß der Implikation auf den Entwurf aus Abschnitt 2.1.1, daß der Designschwerpunkt in der einfachen Bedienung liegen muß, werden dem Benutzer folgende Funktionen angeboten:

- Neue Anfrage stellen
- Anfrage beenden
- Liste der eigenen zur Zeit dargestellten und wartenden Anfragen
- Hilfefunktion

Der Funktionsumfang ist zur Erleichterung der Bedienung auf das Nötigste reduziert. Die weitere Optimierung der Auswahl der Ausgabegeräte ist allein durch Kontextinterpretationen zu realisieren. In einer ausreichend mit Ausgabegeräten ausgestatteten Umgebung ist davon auszugehen, daß alle Anfragen der Benutzer sofort in ihrer

Nähe dargestellt werden können. Dieses erkennt der Anfragesteller intuitiv, so daß keine zusätzliche Rückmeldung bei Darstellungsbeginn notwendig ist. Zu Zeiten außergewöhnlich hoher Aktivität der Benutzer oder außergewöhnlich vieler Benutzer in der lokalen Umgebung können jedoch Engpässe bei der Belegung der Ausgabegeräte auftreten. In diesem Fall müssen Anfragen kurzzeitig in einer Warteschlange verwaltet werden, bis geeignete Ausgabegeräte wieder zur Verfügung stehen. Bei eigenen Anfragen, die sich in der Warteschlange befinden, wird die geschätzte noch zu wartende Zeit bei der oben angeführten Listenfunktion mitangegeben. Zusätzlich zu diesen jederzeit vom Benutzer auswählbaren Funktionen verfügen die Eingabegeräte über eine Alarmfunktion, um den Darstellungsbeginn einer Anfrage mitzuteilen. Dieses ist sowohl nötig bei Anfragen, die in anderen Parzellen als der Anfrageparzelle dargestellt werden, als auch bei einer längeren Verzögerung zwischen Anfragezeitpunkt und Darstellungszeitpunkt. In beiden Fällen ist es für den Benutzer nicht intuitiv ersichtlich, wann und wo seine Anfragen ausgegeben werden, daher muß dieses ihm explizit mitgeteilt werden.

Zur Identifizierung der Benutzer und somit zur Zuordnung der Anfragen wird eine eindeutige Identifikation ihrer Eingabegeräte verwendet. Diese Identifizierung ist jedoch lediglich gebunden an einzelne Anfragen, so daß sie bei neuen Anfragen immer wieder verschieden zugeordnet werden kann. Damit können die Implikationen bzgl. der Sicherheitsaspekte aus Abschnitt 2.1.2 umgesetzt werden. Die Geräte müssen mobil mit dem Intranet verbunden sein und einen Lokalisierungsmechanismus unterstützen. Eine Implementierungmöglichkeit ist in Kapitel 6 vorgestellt.

3.1.2 Controller

Der Controller ist die Verwaltungseinheit des Systems. Er stellt Schnittstellen für die Eingabegeräte zur Verfügung, verarbeitet Kontextinformationen wie die in Abschnitt 3.2.1 dargestellten Lokationsinformationen, bestimmt das Ausgabegerät einer Anfrage mit Hilfe eines Auswahlalgorithmus (vgl. Abschnitt 3.2.2) und beauftragt dieses mit der Mediendarstellung. Dazu verwendet er eine Gerätedatenbank, die Standort, Belegung und Darstellungsmöglichkeiten und -qualitäten der Ausgabegeräte enthält.

3.1.3 Ausgabegeräte

Die Ausgabegeräte stehen dem Controller als Dienstgeber zur Verfügung, die er beauftragt, indem er ihnen die Adresse des von ihnen darzustellenden Medienstroms übergibt. Sie müssen daher in der Lage sein, selbst auf die Daten zuzugreifen. Dieses stellt keinen zu hohen Anspruch dar, da neuere Generationen auch althergebrachter Geräte wie z. B. Radios als Internetradios standardmäßig über die Fähigkeit verfügen, über digitale Netze zu kommunizieren. Jedem Ausgabegerät ist eine Menge von Medientypen aus Tabelle 2.1 zugeordnet, die es darstellen kann. Anhand dieser Mengen sucht der Controller bei einer neuen Anfrage zunächst nach potentiellen Ausgabegeräten, bevor er den Belegungsgrad und die Entfernung zum Anfragesteller berücksichtigt.

Prinzipiell kämen auch die Eingabegeräte als Ausgabegeräte in Frage. Dabei handelt es sich um eine konzeptionelle Entscheidung. Betrachtet man das System schwerpunktmäßig als einen Web-Browser für einen PDA, der sich zur Darstellung nichtkompatibler Medienformate der Umgebung bedient, so sollte der PDA sogar das Standardausgabegerät sein. Dieser Ansatz wird von Beigl u. a. in „The Ubicomp-Browser“ [6] beschrieben. In dieser Arbeit werden die mobilen Eingabegeräte jedoch als reine Steuereinheiten betrachtet, um die Verteilung ihrer Anfragen auf ausnahmslos nichtmobile Ausgabegeräte zu untersuchen.

Direkte Interaktionen eines Benutzers, z. B. Blättern oder Spulen, mit einem Ausgaberät, das gerade eine seiner Anfragen darstellt, sind möglich. Der Benutzer kann die Darstellung entweder auf seinem Eingabegerät beenden oder direkt auf dem Ausgabegerät. Im letzteren Fall schickt das Ausgabegerät eine Nachricht an den Controller.

3.2 Funktionen

3.2.1 Lokalisation

Neben ihrem Belegungsgrad und ihrer technischen Fähigkeit zur Darstellung bestimmter Medienarten spielt die Entfernung der potentiellen Ausgabegeräte zum Benutzer eine entscheidende Rolle dabei, welches Gerät mit der Darstellung beauftragt wird. Ausgabegeräte, die sich in der Nähe des Anfragestellers befinden, sind bevorzugt auszuwählen. In diesem Abschnitt wird definiert, was „in der Nähe“ bedeutet und wie dies im Controller repräsentiert wird.

Bei der Verarbeitung der Lokalisationsinformation wird ein zweistufiges Verfahren angewendet. Für die erste Stufe bleibt zunächst festzustellen, daß weniger der physikalische Abstand eine Rolle spielt als die Sichtbarkeit und Erreichbarkeit der Ausgabegeräte vom Standort des Benutzers. Um diese zu bestimmen, wird die gesamte vom System abgedeckte Umgebung in Parzellen eingeteilt. Diese Aufteilung ist so vorzunehmen, daß jedes Ausgabegerät genau einer Parzelle zugeordnet werden kann, in der es als sichtbar betrachtet wird und von einem Benutzer innerhalb der Parzelle erreicht werden kann, ohne diese zu verlassen. Die Parzellen dürfen beliebige Form haben und eine beliebige Anzahl von Ausgabegeräten enthalten, somit auch leer sein, wie es z. B. bei Fluren der Fall sein kann. Vorzugsweise sollten nichtleere Parzellen konvex sein, damit die Sichtbarkeit aller Ausgabegeräte, die sich normalerweise am Parzellenrand befinden, von allen Punkten innerhalb der Parzelle gewährleistet ist. Größere Räume mit vielen Ausgabegeräten sollten in mehrere Parzellen aufgeteilt

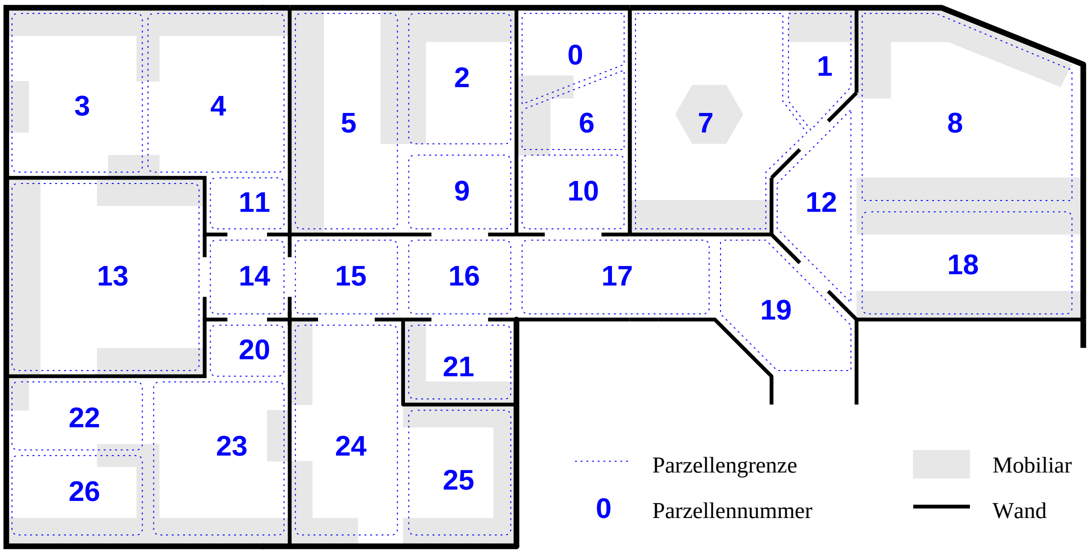

Abbildung 3.1: Aufteilung einer Umgebung in Parzellen

werden. Abbildung 3.1 zeigt eine solche Aufteilung anhand des Grundrisses des Telecooperation Office der Universität Karlsruhe. Die Position der Ausgaberäte und ihre Sichtbarkeitsbereiche, die natürlich eine entscheidende Rolle bei der Festlegung der Parzellen spielen, sind der Übersicht halber weggelassen. Abbildung 3.2 zeigt beispielhaft für ein Büro die Verteilung der Ausgabegeräte mit ihren Ausrichtungen, aus denen die Sichtbarkeitsbereiche abgeleitet werden können. Bei den großen Flachbildschirmen handelt es sich um die schon von Mark Weiser angesprochenen Boards [23], die in einem mit Ubiquitous Computing Hardware ausgestatteten Büro die Rolle von Tafeln, Wandkalendern, Stadtkarten oder auch Bildern übernehmen.

Die Topologie der Umgebung kann mit Hilfe eines ungewichteten Digraphen, dem sogenannten Erreichbarkeitsgraphen, beschrieben werden. Zwei aneinandergrenzende Parzellen werden genau dann mit einer Kante verbunden, wenn ein Benutzer zwischen diesen verkehren kann, sei es offen innerhalb eines Raumes oder durch eine Tür. Abbildung 3.3 zeigt den zur Abbildung 3.1 gehörigen Erreichbarkeitsgraphen. Die Position des Anfragestellers ist somit auch durch seine Aufenthaltsparzelle für die erste Lokalisierungstufe genügend genau bestimmt. Wie in Abschnitt 2.2.1 betont, ist diese grobe Granularität mit heutigen Lokalisierungstechniken zuverlässig ortbar. Die Distanz zwischen dem Anfragesteller und den potentiellen Ausgabegeräten wird somit in ganzen Zahlen angegeben, die die Entfernung in Parzellen beschreiben. So haben alle Ausgabegeräte in der Anfrageparzelle die Entfernung Null vom Anfragesteller. Damit Entfernungen über mehrere Parzellen realistisch miteinander verglichen werden können, ist es vorteilhaft, wenn die Parzellengrößen nicht zu sehr voneinander abweichen. So sollten z. B. lange Gänge durch Parzellenketten modelliert werden. Bei der Modellierung der Umgebung durch die Einteilung in Parzellen sollte stets bedacht werden, daß diese einen wesentlichen Effekt auf die Güte des arbeitenden Systems hat.

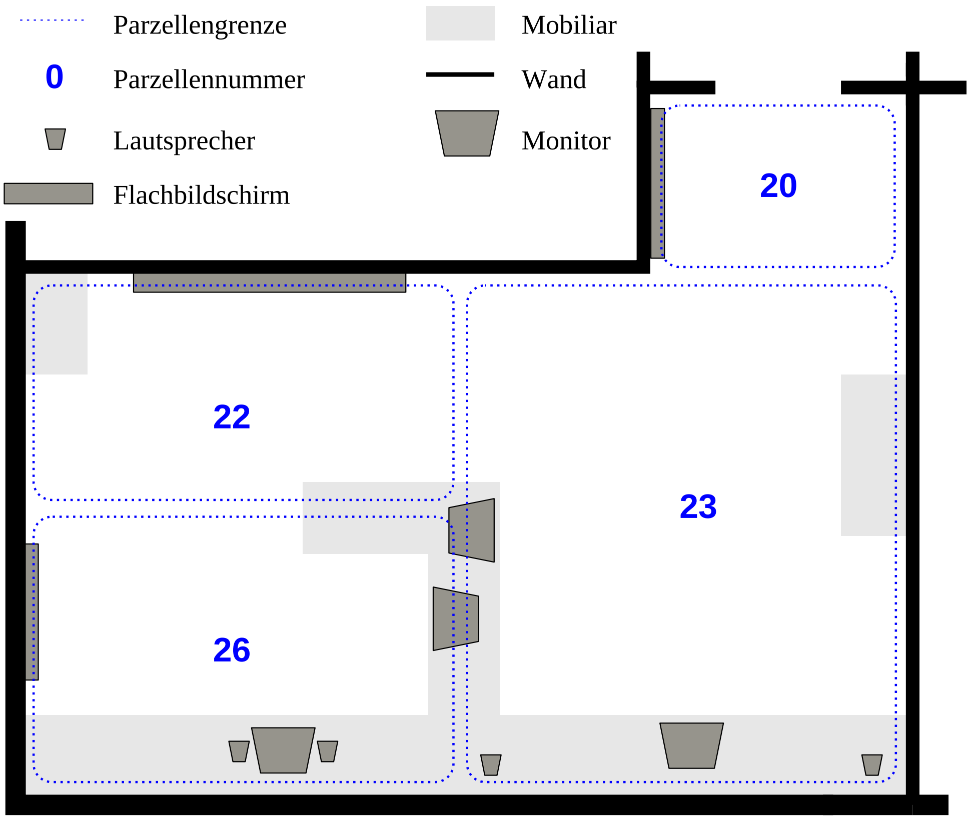

Abbildung 3.2: Büroparzellierung mit Ausgabegeräten

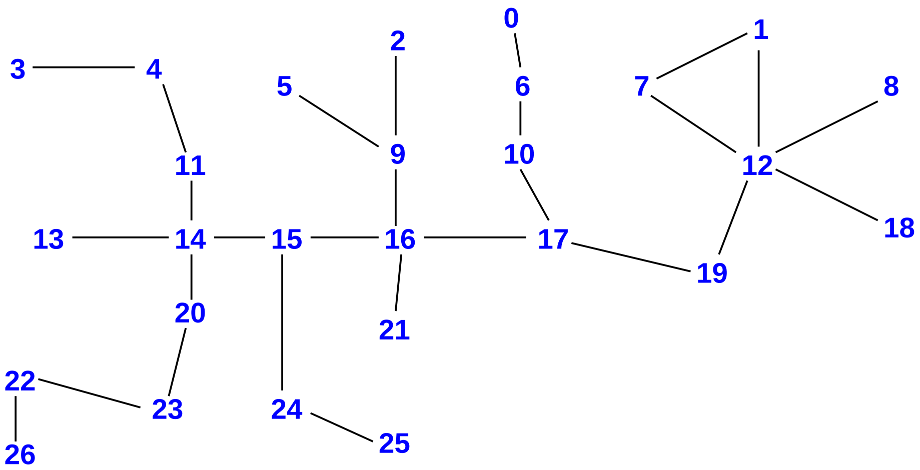

Abbildung 3.3: Erreichbarkeitsgraph zu Abbildung 3.1

Bei der Modellierung von Umgebungen, die größere Bereiche beinhalten, in denen keinerlei Interaktionen möglich sind, sei es Anfragen zu stellen oder Medienströme auszugeben, sollten gewichtete Kanten verwendet werden. Hierbei gibt das Gewicht die Entfernung zwischen zwei Parzellen an. So lassen sich zum Beispiel Wege zwischen Gebäuden oder durch Treppenhäuser zwischen Stockwerken beschreiben. Der kürzeste Weg zu Ausgabegeräten in anderen Parzellen läßt sich dann im zugehörigen Distanzgraphen mittels Dijkstras Algorithmus [14] bestimmen.

Wenn sich viele Ausgaberäte mit sich überschneidenden Fähigkeiten innerhalb einer Parzelle befinden, ist eine genauere Positionsverarbeitung notwendig, um das wirklich naheliegendste zu finden. Hier wird als zweite Stufe der Lokalisation zusätzlich die physikalische Distanz vom Benutzer zu den potentiellen Ausgabegeräten innerhalb der Parzelle betrachtet. Dazu werden lokale Positionskoordinaten verwendet, die die Position innerhalb der Parzelle beschreiben. Daher werden alle Lokalisationsinformationen sowohl der Benutzer als auch der Ausgabegeräte vom HermiUC-System in Tupelform `(Parzelle, Positionskoordinaten)` verarbeitet.

3.2.2 Auswahl der Ausgabegeräte

Wie bereits in Abschnitt 1.2 angesprochen wurde, wird es angestrebt, daß der Controller das Ausgabegerät bestimmt, welches der Benutzer selber als das geeigneteste anerkennt. Daher sollten die Kriterien, die dem Auswahlalgorithmus zu Grunde liegen, einfach nachzuvollziehen sein. Als Grundkriterien werden verwendet:

Technische Eignung: Es werden nur Ausgabegeräte betrachtet die technisch zur Darstellung des Medientyps der Anfrage in der Lage sind.

Belegungszustand: Die Ausgabegeräte dürfen zur Zeit nicht belegt sein.

Erreichbarkeit: Potentielle Ausgabegeräte müssen sich in der Nähe des Benutzers (vgl. Abschnitt 3.2.1) befinden.

Typspezialisierung: Es werden bevorzugt die Ausgabegeräte ausgewählt, die die geringste Anzahl an unterschiedlichen Medientypen unterstützen (vgl. Abschnitt 3.1.3). Dieses Vorgehen minimiert für nachfolgende Anfragen die Wahrscheinlichkeit, daß sie nicht dargestellt werden können, obwohl noch andere Geräte frei sind, diese jedoch nicht ihren Medientyp unterstützen. So wird z. B. für Audioanfragen ein Radio gegenüber einem Computer mit Lautsprechern bevorzugt, damit der Computer für eventuell noch eintreffende Anfragen von Videoströmen weiterhin zur Verfügung steht.

Darstellungqualität: Ausgabegeräte mit höherer Darstellungsqualität sind vorzuziehen. Diese kann z. B. bei Videoausgabegeräte als Bildschirmauflösung oder -diagonale verglichen werden.

Obwohl es zunächst nahe läge, haben bereits aktive Anfragen eines Benutzers aus folgenden Gründen keinen Einfluß bei der Auswahl weiterer Ausgabegeräte. Bei statischen Medientypen ist eine parallele Ausgabe ohnehin unproblematisch. Mehrfache

Anfragen nichtstatischer Typen sollten nacheinander erfolgen, wenn sie nacheinander dargestellt werden sollen. Bei paralleler Anfragestellung erfolgt auch eine parallele Ausgabe, sofern dieses die Infrastruktur zuläßt. Parallele Anfragen in unterschiedlichen Parzellen sind vom Benutzer häufig durchaus gewollt. So läuft z. B. im allgemeinen das Radio im eigenen Büro weiter und die momentane Anwendung bleibt auf dem Rechner, wenn man kurz einen Kollegen aufsucht, um mit ihm einen Bericht zu besprechen, wie dies bereits im Anwendungsszenarium in Abschnitt 1.3 dargestellt wurde.

3.2.2.1 Auswahlalgorithmus

Anfragen von Medienströmen werden vom Controller mit Hilfe einer Warteschlange verwaltet, in die zunächst auch alle neuen Anfragen aufgenommen werden. Bei jedem Eintreffen einer neuen Anfrage oder Beenden einer Darstellung wird folgender Algorithmus für alle Anfragen in der Warteschlange in der Reihenfolge ihres Anfragezeitpunktes startend mit der ältesten durchlaufen, bis eine neue Darstellung in Auftrag gegeben wird oder die Warteschlange komplett durchsucht wurde. Bei dieser Vorgehensweise werden Anfragen, die um die gleichen Ausgabegeräte konkurrieren, automatisch immer nach der first-come-first-serve-Methode (FCFS) zur Darstellung gebracht. Da hinter den Anfragen Menschen stehen, die sich bei für sie unübersichtlichen Ordnungsmethoden schnell benachteiligt fühlen, ist dieses die einzige geeignete Vorgehensweise. Im folgenden Pseudokode bezeichnen TA, PTA, etc. Mengen, in denen durch immer weitere Auswahl das Ausgabegerät (AG) gesucht wird. Die Notation ist so zu verstehen, daß PTA eine Untermenge von TA darstellt.

Der Algorithmus arbeitet bei der Bestimmung des Ausgabegerätes in einem mehrstufigen Verfahren. Im Hauptprogramm (Algorithmus 3.1) wird zunächst bestimmt, ob die untersuchte Anfrage zum aktuellen Zeitpunkt erfüllt werden kann, worauf im positiven Fall mit Hilfe der Prozedur fpta-Auswahl (Algorithmus 3.2) das passende Ausgabegerät bestimmt wird.

Das Hautprogramm beginnt mit einer Bestimmung der Menge aller technisch geeigneten Ausgabegeräte in der gesamten Umgebung. Im nichtleeren Fall ist die Darstellungsmöglichkeit der Anfrage zu einem späteren Zeitpunkt gesichert. Dann werden rekursiv im Erreichbarkeitsgraphen beginnend mit der Anfrageparzelle die parzellenmäßig naheliegendsten Geräte aus dieser Menge gesucht. Die Ergebnismenge ist als PTA-Menge bezeichnet. Daraus wird die Untermenge der momentan freien Ausgabegeräte bestimmt und im nichtleeren Fall, also wenn eine sofortige Darstellung möglich ist, an die fpta-Auswahl Prozedur übergeben. Wenn keines der naheliegenden technisch geeigneten Ausgabegeräte frei ist, wird das nächstliegende freie geeignete Gerät bestimmt. Nun muß die Entscheidung getroffen werden, den Anfragesteller zu diesem entfernteren Ausgabegerät zu schicken, oder ihn warten zu lassen, bis ein Ausgabegerät in seiner Nähe frei geworden ist. Seine Anfrage wird sofort auf dem entfernteren Gerät dargestellt, wenn mindestens eines der beiden folgenden Kriterien erfüllt ist:

Algorithmus 3.1 Hauptprogramm

```
if TA der für den Medientyp geeigneten AGs nichtleer then
  while PTA der AGs in der Anfragenparzelle leer do
    erweitere PTA um AGs in den Nachbarparzellen
  end while
  if FPTA der freien AGs nichtleer then
    fpta-Auswahl
  else
    repeat
      erweitere FPTA um freie AGs in den Nachbarparzellen
    until not (FPTA leer and noch Nachbarparzellen da)
    if ( FPTA nichtleer ) and ( [ geschätzte Wartezeit auf AG in Anfrageparzelle
    − Entfernungszeit > Trägheitsschwelle ] or [ bisherige Wartezeit der Anfrage
    ≥ Ungeduldsschwelle ] ) then
      fpta-Auswahl
    end if
  end if
else
  keine Ausgabe möglich, entferne Anfrage aus Warteschlange
end if
```

Algorithmus 3.2 Prozedur fpta-Auswahl

```
if FPTA einelementig then
  wähle FPTA
else if MFPTA der meisttypspezialisierten AGs einelementig then
  wähle dieses
else if MFPTA in Anfrageparzelle then
  wähle physikalisch naheliegendstes AG
else
  wähle auflösungsstärkstes AG
end if
```

1. Die Differenz zwischen der geschätzten Wartezeit auf das Freiwerden eines naheliegenden Ausgabegerätes und der Zeit, die der Benutzer benötigt, sich zu dem Entfernteren zu begeben, ist größer als eine festgesetzte Trägheitsschwelle. Diese Trägheitsschwelle gibt einen Zeitwert an, der bestimmt, ab welcher Verkürzung der Wartezeit man bereit ist, sich von seinem derzeitigen Platz wegzubewegen. Hierbei ist zu beachten, daß, wenn Ausgabegeräte von Medientypen mit begrenzter Länge belegt sind, die Zeit bekannt ist, ab welcher das Ausgabegerät spätestens wieder zur Verfügung steht, so daß nicht geschätzt werden muß. Die Schätzung erfolgt durch automatische Analyse früherer Anfragen desselben Medientyps. Wenn nicht nur signifikante Unterschiede der Betrachtungslängen bzgl. des Typs, sondern auch in Hinblick auf verschiedene Parzellen, Tageszeiten oder Benutzer ermittelt werden, sollten auch diese berücksichtigt werden. Das führt zu einer Lernfähigkeit des Systems, die sich in der Verbesserung von Voraussagen des Darstellungsbeginns und somit auch einer verbesserten Verteilung der Anfragen widerspiegelt. Bei der Schätzung des Zeitpunktes, zu dem ein momentan belegtes Ausgaberät wieder frei wird, ist Folgendes zu beachten. Die Berücksichtigung der schon vergangenen Darstellungszeit der momentan aktiven Anfrage ist nur unter der Voraussetzung sinnvoll, daß die Verteilung der Darstellungslänge des Medientyps der aktiven Anfrage nicht gedächtnislos ist. Andernfalls ist die Wahrscheinlichkeit, daß man noch weitere t Sekunden bis zum Darstellungsstopp warten muß, immer gleich groß, unabhängig wie lange der Medienstrom schon dargestellt wird. Hierauf wird ausführlicher in Kapitel 4 eingegangen.

2. Die bereits vergangene Wartezeit hat eine festgesetzte Ungeduldsschwelle überschritten. Dieses Kriterium stellt sicher, daß jede Anfrage nach endlicher Zeit dargestellt wird, sofern irgendwo noch Ressourcen zur Verfügung stehen. Simulationsläufe (vgl. Kapitel 5) ohne diese Bedingung haben gezeigt, daß es andernfalls zu extrem langen Wartezeiten innerhalb einer Parzellen kommen kann, wenn dort zeitlich unbegrenzte Medientypen deutlich länger als die durchschnittliche Betrachtungszeit, die zur Schätzung der Wartezeit bei Kriterium 1 verwendet wird, angeschaut werden. Zu dem exakten Zeitpunkt, an dem diese Ungeduldsschwelle überschritten wird, wird eine sofortige Kontrolle auf Ausgabemöglichkeit der entsprechenden Anfrage vorgenommen. Dieses wird zusätzliche zu dem am Beginn dieses Abschnitts vorgestellten Vorgehen anhand der Warteschlange unternommen, um sicherzustellen, daß keine Verzögerung auftritt.

Beim Aufruf der fpta-Prozedur aus dem Hauptprogramm wird die Menge der technisch geeigneten, nach Erreichbarkeitsgraph naheliegendsten, freien Ausgabegeräte übergeben. Ist diese einelementig, ist keine weitere Auswahl nötig. Andernfalls wird die Untermenge der meisttypspezialisierten Ausgabegeräte betrachtet. Besteht diese aus Ausgabegeräten aus der Anfrageparzelle, so wird mit Hilfe der zweiten Lokalisationsstufe das physikalisch Naheliegendste gewählt. Andernfalls ist nicht dieses

Kriterium anzuwenden, da es nur lokal innerhalb von Parzellen definiert ist, sondern es wird das auflösungsstärkste Geräte gewählt.

3.2.3 Medienausgabe

Die Darstellung der Ausgabeströme wird komplett von den Ausgabegeräten selbst erledigt. Sie erhalten vom Controller keine Mediendaten, sondern lediglich ihre Adresse. Bei einer Implementierung ist daher zu berücksichtigen, daß genug Bandbreite für die Medienströme zur Verfügung steht. Weiterhin gibt es zu beachten, daß derselbe Medientyp in verschiedenen Formaten gespeichert sein kann. So können z. B. Filme im PAL oder als MPEG zur Verfügung gestellt sein. Ob automatische Formatkonvertierungen oder Formatunterstützungsinformationen in der Gerätedatenbank zur Lösung verwendet müssen, hängt von den vorhandenen Ausgabegeräten und daher einer speziellen Implementierung ab.

3.3 Kommunikationsablauf

In diesem Abschnitt wird das Zusammenwirken aller bisher vorgestellten Komponenten beschrieben. Abbildung 3.4 zeigt die Arbeitsweise des Systems anhand eines Flußdiagramms. Es ist der Kommunikationsablauf, der bei jeder neuen Anfrage durchlaufen wird, dargestellt. Hierbei ist Parallelität, die bei mehreren gleichzeitigen Anfragen in Teilbereichen zwangsläufig auftritt, der Übersicht halber nicht berücksichtigt. Zur Vermeidung von Inkonsistenzen existiert nur eine Instanz, die die Verarbeitung der Warteschlange und darin eingegliedert den Auswahlalgorithmus übernimmt. Es ist jedoch parallele Kommunikation zwischen den Eingabegeräten, dem Controller und den Ausgabegeräten bzgl. verschiedener Anfragen notwendig. Eingabegeräte sind beispielhaft als PDAs bezeichnet, dieses ist jedoch nicht limitierend zu sehen.

Anmerkungen zu Abbildung 3.4

(*1) Wann bei Darstellungsbeginn eine Meldung an den PDA erforderlich ist, ist in Abschnitt 3.1.1 erklärt.

(*2) Wenn das Herunterladen des Medienstroms vom Ausgabegerät nicht möglich ist, oder die Daten fehlerhaft sind, so daß keine Darstellung erfolgen kann, gibt das Ausgabegerät selbst eine Fehlermeldung aus. Die Bestätigung dieser Meldung wird wie ein Befehl zur Beendigung einer Anfrage abgearbeitet, so daß keine zusätzlichen Fehlerbehandlungsroutinen nötig sind.

(*3) Das Beenden der Darstellung einer Anfrage kann auf drei verschiedene Weisen erfolgen. Erstens, bei Medientypen mit begrenzter Länge wie z. B. Filmen wird die

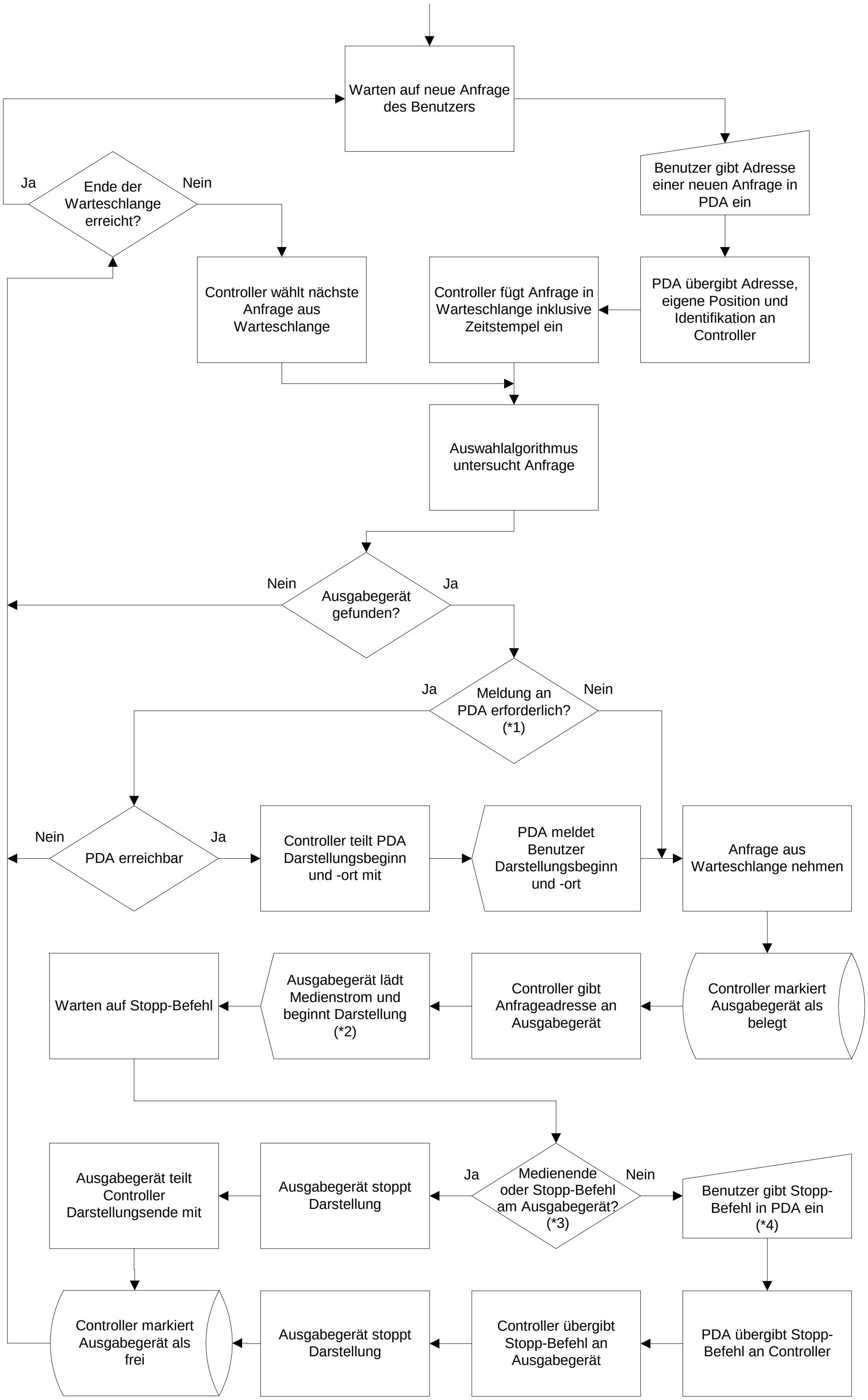

Abbildung 3.4: Flußdiagramm zum Kommunikationsablauf

Darstellung am Ende automatisch gestoppt und das Ausgabegerät teilt dies dem Controller mit, so daß in diesem Fall keine Aktion des Benutzers verlangt ist. Zweitens, der Benutzer wählt den Stopp-Befehl auf seinem PDA, in welchem Fall die Weiterleitung des Befehls über den Controller erfolgt. Drittens, der Benutzer befiehlt nicht am PDA, sondern direkt am Ausgabegerät den Darstellungsstopp. Hier nun stoppt das Ausgabegerät die Darstellung sofort und geht dann wie im ersten Fall vor.

(*4) Der Benutzer kann eine gestellte Anfrage zu jedem Zeitpunkt auf seinem PDA beenden. Wenn die Darstellung noch nicht begonnen hat, wird die entsprechende Anfrage aus der Warteschlange gelöscht.

4. Stochastisches Modell

4.1 Motivation der Modellierung und Simulation

Eine Implementierung größeren Maßstabs des in Kapitel 3 entworfenen HermiUC-Systems ist momentan nur mit erheblichen Zeit- und Kostenaufwand realisierbar und übersteigt den Umfang dieser Arbeit. Dieses liegt hauptsächlich im Fehlen der notwendigen ubiquitären Infrastruktur, auf der das System aufbaut, begründet. Dabei wären vor allem eine zuverlässige Positionserkennung und eine ausreichende Anzahl mobiler persönlicher Geräte mit drahtloser Verbindung zum Intranet und ausreichender Batterielebensdauer zu nennen. Ebenso ist eine theoretische mathematische Analyse mit Methoden aus dem Operations Research aufgrund der Komplexität des Systems offenbar nur in einfachen Spezialfällen möglich. Um dennoch gewisse Aussagen über die Leistungsfähigkeit des Systems treffen zu können und eine Optimierung vorzunehmen, muß auf eine Simulation zurückgegriffen werden. Das in diesem Kapitel entworfene stochastische Modell dient als Grundlage der in Kapitel 5 beschriebenen Simulation. Mit Hilfe der mathematischen Analyse und parallelen Simulation der Szenarien in Abschnitt 5.3 wird durch Vergleich der Ergebnisse die Korrektheit der Simulation validiert. Da bisher keine Ubiquitous Computing Umgebungen der hier beschriebenen Art und Größe aufgebaut worden sind, liegen naturgemäß keinerlei reale Daten über das Benutzerverhalten darin vor. Daher müssen alle zufallsbedingten Prozesse mit Hilfe von zu bestimmenden Verteilungsannahmen simuliert werden. Sowohl die Zeit als auch der Ort, der Medientyp und die Darstellungsdauer einer Anfrage können als zufällig betrachtet werden. Nach einer Implementierung wäre eine Untersuchung des Benutzerverhaltens interessant, um zu überprüfen, ob die hier gemachten Annahmen realistisch gewesen sind.

4.2 Modellvereinfachungen

Das Modell basiert auf dem implementierungsunabhängigen Aufbau aus Kapitel 3. Es wird daher ausschließlich die Anwendungsschicht des ISO/OSI-Schichtenmodells berücksichtigt. Ausreichende Prozessorkapazitäten, ständige Kommunikationsmöglichkeit ohne Zeitverlust zwischen allen Komponenten und genügende Bandbreite für Systemkommunikation und Medienströme werden vorausgesetzt.

Da das Ziel der Simulation die Optimierung der Verteilung der Ausgabeströme ist, setzt auch das Modell an genau dieser Stelle an. Es wird nicht die gesamte in Abschnitt 3.3 beschriebenen Arbeitsweise des Systems modelliert, sondern sich auf die für die Verteilung entscheidenden Komponenten beschränkt. Das bedeutet, daß Anfragen der Benutzer simuliert werden, jedoch nicht die Benutzer selber. Dieses stellt keine Einschränkung dar, weil bei der Auswahl der Ausgabegeräte nach Abschnitt 3.2.2 die Identität des Anfragestellers keinen Einfluß nimmt. Bei einer Implementierung dagegen ist die Identifizierung der Anfragesteller nötig, um die in Abschnitt 3.1.1 beschriebenen Funktionen, wie z. B. die Rückmeldung bei Darstellungsbeginn einer ihrer Anfragen, zu Verfügung zu stellen. Bei der Analyse der Verteilung der Ausgabeströme spielen jedoch nur zwei der dort beschriebenen Funktionen eine Rolle, nämlich das Stellen neuer Anfragen und das Beenden von Anfragen. Es ist jedoch ebenfalls unerheblich, auf welche Weise diese Funktionen vorgenommen werden, so daß es beim Beenden einer Anfrage durch den Anfragesteller nicht unterschieden wird, ob dieses am PDA oder am Ausgabegerät direkt befohlen wurde. Der allgemeinere Unterschied zwischen dem automatischen Darstellungsstopp bei Medientypen mit begrenzter Länge und dem Stopp auf Befehl eines Benutzer muß jedoch schon daher berücksichtigt werden, als daß der Controller beim Auswahlalgorithmus im Hauptprogramm die Wartezeit auf ein freies Ausgabegerät in der Anfrageparzelle berücksichtigt. Dabei weiß er bei Medientypen mit begrenzter Länge, wie lange das Ausgabegerät höchstens noch belegt bleibt. Bei Medientypen mit unbegrenzter Länge, die nur durch den Benutzer beendet werden, muß er dagegen mit Schätzwerten rechnen. Die damit verbundene Analyse und Lernfähigkeit, die in Abschnitt 3.2.2.1 angesprochen ist, wird im Modell jedoch nicht explizit umgesetzt. Da hier die Darstellungslängen mittels stochastischer Verteilungen bestimmt werden, stehen hier als Schätzwerte direkt die Erwartungswerte der entsprechenden Verteilungen zur Verfügung.

Bei den Lokalisationsinformationen wird eine Vereinfachung gegenüber der Realität vorgenommen. Zur Verteilung der Lokalisationsinformation der ersten Stufe, also der Parzelle, in der eine Anfrage gestellt wird, läßt sich ein sinnvolles stochastisches Modell entwickeln. Zur Simulation der zweiten Stufe, der physikalischen Position innerhalb einer Parzelle, wäre jedoch eine getrennte detailierte Untersuchung jeder zu simulierenden Parzelle notwendig, da die möglichen und wahrscheinlichen Positionen der Benutzer dort nicht nur stark von der Position der Ausgabegeräte, sondern auch von der Form des Raumes, vom Mobiliar, von der Anzahl der Menschen im Raum und sogar von den Umgangsformen der Mitarbeiter untereinander abhängen. Man bewegt sich z. B. in einem leeren Raum anders, als wenn man Rücksicht auf jemanden nimmt oder sich auch nur beobachtet fühlt. Ebenso gibt es Menschen, die sofort bis zum Schreibtisch von jemanden treten, wenn sie etwas besprechen wollen, andere jedoch bewahren zunächst mehr Abstand, oder setzten sich auf einen dafür vorgesehenen Besucherstuhl. Unter diesen Umständen eine stochastische Verteilung bestimmen zu wollen, die die Positionkoordinaten einer Anfrage innerhalb einer Parzelle beschreibt, ist nicht sinnvoll. Daher wird beim Modell die zweite Stufe der Lokalisation nicht berücksichtigt. Dieses hat jedoch nur sehr geringen Einfluß auf die Ergebnisse der Simulation, da es im Auswahlalgorithmus erst als letztes Kriterium verwendet wird, um eindeutig ein Ausgabegerät innerhalb einer Parzelle zu bestimmen, wenn alle anderen Kriterien vorher noch zu keiner eindeutigen Wahl geführt haben, und dies auch nur in dem Fall, daß die Ausgabeparzelle gleich der Anfrageparzelle ist.

Ebenso verhält es sich mit der Darstellungsqualität. Dieses Kriterium wird an Stelle der Lokalisation der zweiten Stufe eingesetzt, wenn sich Anfrageparzelle und Ausgabeparzelle unterscheiden. Beide Kriterien bestimmen lediglich, falls mehrere potentielle Ausgabegeräte mit dergleichen Quantität an darstellbaren Medientypen für eine Anfrage in einer Parzelle vorhanden sein sollten, welches als erstes belegt wird. Bei einer nachfolgenden Anfrage gleichen Medientyps wird dann aus den Verbliebenen gewählt. So kommt es durch diese beiden Kriterien normalerweise nur zu einer Vertauschung der Wahl der Ausgabegeräte innerhalb einer Parzelle gegenüber einer anders vorgenommen Auswahl. Da dieses jedoch keinen Einfluß auf den Belegungsgrad innerhalb einer Parzelle hat und weder die Verteilung nachfolgender Anfragen auf die Parzellen noch die Wartezeit derselben beeinflußt, wird diese Vereinfachung gemacht. Sollte bei der Simulation einer dieser seltenen Fälle auftreten, so werden die Ausgabegeräte in der Reihenfolge ihres Eintrags in der Datenbank gewählt.

4.3 Modellbeschreibung

4.3.1 Systemmodell

Bei diesem Modell handelt es sich um ein Warteschlangensystem. Abbildung 4.1 zeigt seine Struktur. Da jederzeit Ankünfte neuer Anfragen der Benutzer in der Quelle entstehen können, handelt es sich um ein offenes Warteschlangennetz. Alle neuen Anfragen werden zunächst in die Warteschlange des Controllers eingereit und dann von dort mit Hilfe des Auswahlalgorithmus auf die Ausgabegeräte verteilt. Anschließend verlassen sie wieder das Netzwerk. Es wird in allen Szenarien von einem unendlich großen Warteraum ausgegangen. Anfragen, für die kein technisch geeignetes Ausgabegerät vorhanden ist, werden direkt vom Controller aus der Warteschlange gelöscht. Da alle Ausgabegeräte als Dienstgeber fungieren, die vom Controller Aufträge erhalten, liegt ein Multi-Server-Warteschlangennetz vor. Bei der Bearbeitung der Aufträge durch die Ausgabegeräte kommt ausschließlich eine unterbrechungsfreie Bedienung zur Anwendung.

Definition 4.1. Für eine positive Zahl λ heißt das zur Dichte

$$f(x) = \begin{cases} 0, & x < 0 \\ \lambda \cdot e^{-\lambda x}, & x \geq 0 \end{cases} \tag{4.1}$$

korrespondierende Wahrscheinlichkeitsmaß *Exponentialverteilung mit Parameter* λ und wird mit $Exp(\lambda)$ bezeichnet.

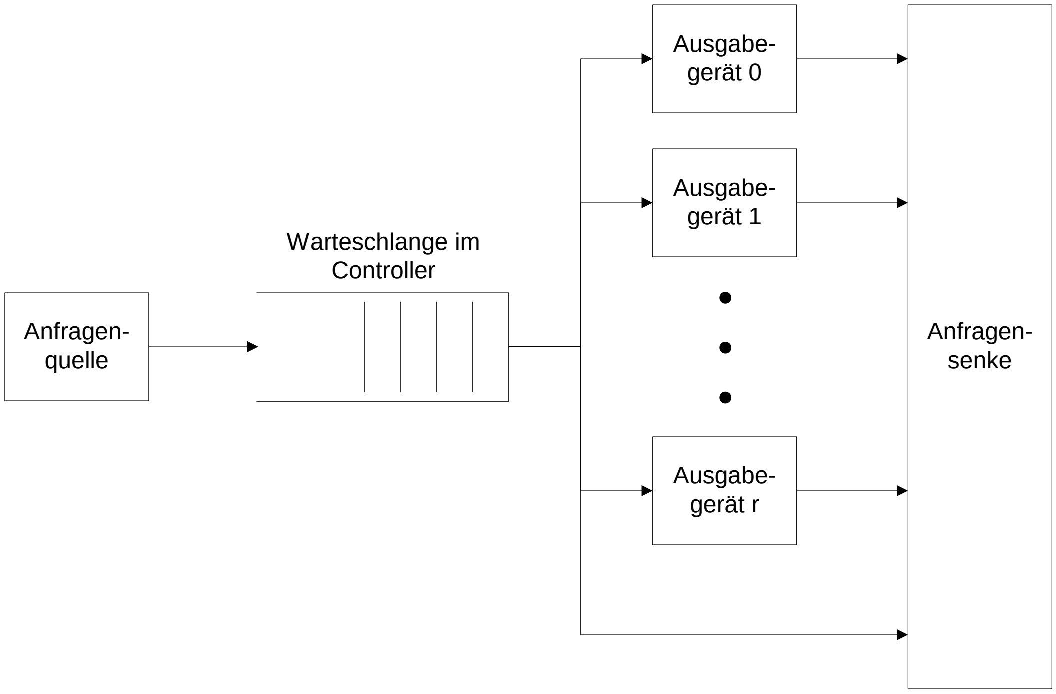

Abbildung 4.1: Struktur des stochastischen Modells

Ist Y exponentialverteilt, so gilt

$$P(Y > t + s \mid Y > s) = P(Y > t), \quad s, t \geq 0. \tag{4.2}$$

Diese Eigenschaft heißt Gedächtnislosigkeit und gilt nur bei der Exponentialverteilung.

Frühere, nicht mehr aktive Anfragen spielen bei der Verteilung neuer Anfragen durch den Auswahlalgorithmus keine Rolle. Dennoch besitzt der Systemzustand im Allgemeinen nicht die Markowsche Eigenschaft, daß der zukünftige Zustand nur vom aktuellen und nicht von früheren Zuständen abhängt. Eine Ausnahme erhält man, wenn man sowohl zur Erzeugung der Darstellungslängen als auch der Anfragezeitpunkte gedächtnislose Verteilungen, d. h. exponentialverteilte Darstellungslängen und Zwischenankunftszeiten der Anfragen, wählt und die Ungeduldsschwelle unberücksichtigt läßt. Wählt man andere Verteilungen, hängt der zukünftige Systemzustand nicht nur vom aktuellen sondern auch von der Zeit ab, die schon bei den momentan dargestellten Anfragen seit Darstellungsbeginn vergangen ist, sowie der verstrichenen Zeit seit der letzten Anfrage.

4.3.2 Anfragenverteilung

Bei den Anfragen im hier dargestellten Modell handelt es sich um die Kunden in der klassischen Warteschlangentheorie. Sie erscheinen nach bestimmten Verteilungsannahmen, werden in die Wartschlange aufgenommen und verbleiben dort, bis sie von einem Ausgabegerät bedient, d. h. dargestellt, werden. Darauf verlassen sie das Netzwerk wieder. Zu jeder Anfrage gehören somit bei ihrer Erzeugung folgende Daten: der Zeitpunkt, zu dem sie gestellt wird, die Parzelle, aus der sie stammt, ihr Medientyp und ihre Darstellungslänge. Dabei ist die Darstellungslänge allein vom Medientyp abhängig. Die Verteilung der Darstellungslängen ist in Abschnitt 4.3.3 angegeben. Es gibt daher verschiedene Klassen von Anfragen. Nur Anfragen, die in Medientyp und Anfrageparzelle übereinstimmen, konkurrieren direkt um dieselben Ausgabegeräte und sind deshalb einer Klasse zuzuordnen.

Zunächst wird die Erzeugung der Anfragezeitpunkte betrachtet. Das aufgrund seiner angenehmen Eigenschaften bestuntersuchteste und bei Netzwerksimulationen überwiegend eingesetzte Modell ist der Poisson-Prozeß. Dieser wird auch hier in einer geeigneten Form verwendet.

Definition 4.2. Ein *(homogener) Poisson-Prozeß mit Ankunftsrate* $\lambda > 0$ ist eine Folge von Zufallszeiten $\{S_n,\ n \geq 1\}$ mit $0 < S_1 < S_2 < \ldots$ derart, daß die Zwischenankunftszeiten $S_{n+1} - S_n$ für $n \geq 0$ (mit $S_0 := 0$) unabhängig und exponentialverteilt mit Parameter λ sind.

Der Zählprozeß

$$N(t) = \max\{n \geq 0 \,|\, S_n \leq t\}, \quad t \geq 0 \tag{4.3}$$

wird dann ebenfalls (homogener) Poisson-Prozeß mit Ankunftsrate λ genannt.

Per Definition tragen die Anfragen nichtidentische Zeitstempel. Dieses stellt jedoch keine Beschränkung der Allgemeinheit dar, da verschiedene Benutzer zwar theoretisch versuchen können, zum selben Zeitpunkt Anfragen zu stellen, der Controller diese jedoch nacheinander in die Warteschlange aufnimmt und ihnen somit verschiedene Zeitstempel zuweist.

Die Annahme, daß die Ankunftsrate der Anfragen zeitunabhängig ist, ist unrealistisch. Menschen schlafen, nehmen Mahlzeiten zu sich, gehen zur Arbeit. Bei der Simulation einer Büroumgebung an Arbeitstagen ist von einem 24-Stunden Rhythmus auszugehen. Die ersten Anfragen treffen morgens ein, wenn die Mitarbeiter zur Arbeit erscheinen. Am Vormittag und Nachmittag ist mit einer etwa gleichbleibenden Ankunftsrate zu rechnen. Zur Mittagsessenzeit ist jedoch von einer vorübergehenden Abschwächung auszugehen. Wenn die Mitarbeiter das Büro wieder verlassen, geht die Ankunftsrate wieder auf Null zurück. Dieses läßt sich mit Hilfe einer Intensitätsfunktion $\lambda(t)$ modellieren, die die Ankunftsrate zeitabhängig beschreibt. Diese Funktion ist somit als Verallgemeinerung des konstanten Parameters λ in Definition 4.2 zu betrachten. Durch diese Verallgemeinerung gelangt man zum inhomogenen Poisson-Prozeß.

Ein zufälliges Zählmaß N auf $[0, \infty)$ ist eine Familie $\{N(B)\}$ von $\bar{\mathbb{Z}}^+$-wertigen Zufallsvariablen, wobei B das System aller Borelschen Teilmengen von $\mathbb{R}^+$ durchläuft, also $B \in \mathcal{B}^+$. Dabei ist (für jede Realisierung des Zufalls) $B \mapsto N(B)$ ein lokales endliches Maß mit der zusätzlichen Eigenschaft $N(\{t\}) \leq 1,\ t \geq 0$. Ein zufälliges Maß entspricht in eineindeutiger Weise einer Folge $\{S_n,\ n \geq 1\}$ von $(0, \infty]$-wertigen Zufallsvariablen mit $S_n \leq S_{n+1},\ n \in \mathbb{N}$, und $S_n < S_{n+1}$ falls $S_n < \infty$. Dabei gilt $\lim_{n\to\infty} S_n = \infty$. Der Zusammenhang ist durch $N(B) = \sum_{n=1}^{\infty} \mathbf{1}_{\{S_n \in B\}}$ gegeben.

Definition 4.3. Gegeben sei eine lokal integrierbare meßbare Funktion λ. Ein Zählmaß N oder die korrespondierende Folge von Zufallszeiten $\{S_n,\ n \geq 1\}$ heißt *(inhomogener) Poisson-Prozeß mit Intensitätsfunktion* λ, falls die beiden folgenden Eigenschaften erfüllt sind:

(i) $N(B_1), \ldots, N(B_m)$ sind stochastisch unabhängig für alle $m \in \mathbb{N}$ und alle paarweise disjunkten Borelschen Mengen $B_1, \ldots, B_m$.

(ii) Für alle Borelschen Mengen B gilt

$$P(N(B) = j) = \frac{\left(\int_B \lambda(s)ds\right)^j}{j!} e^{-\int_B \lambda(s)ds}, \quad j = 0, 1, \ldots \tag{4.4}$$

Im Falle einer konstanten Intensitätsfunktion $\lambda(t) \equiv \lambda$ sind Definition 4.3 und Definition 4.2 äquivalent.

Die Anfragezeitpunkte werden durch einen inhomogenen Poisson-Prozeß modelliert. Die jeweils verwendete Verteilungsfunktion $\lambda(t)$ ist bei den Simulationsszenarien (vgl. Abschnitte 5.3 und 5.4) angegeben. $\lambda(t)$ muß dem Szenarium angepaßt sein, kann jedoch zur Analyse der Stabilität der Simulation leicht variiert werden.

Neben dem Zeitstempel S_n, der als Identifikationschlüssel der Anfragen dient, tragen sie eine Marke $X_n \in \mathbb{X}$ mit $X_n = (\text{Medientyp}, \text{Anfrageparzelle})$. Der Markenraum ergibt sich somit zu $\mathbb{X} = \{1, \ldots, 7\} \times \{0, \ldots, k-1\}$, wobei k der Anzahl der in der Umgebung vorhandenen Parzellen entspricht. Diese Marken dienen der Zuordnung der Anfragen zu den verschiedenen Klassen. Somit bezeichnet $(S_1, (3, 5))$ eine Anfrage, mit der zum Zeitpunkt S_1 in der 5. Parzelle ein Dokument mit dem Medientyp 3 angefordert wurde. Für eine Simulation müssen Wahrscheinlichkeiten für jede Marke zur Erzeugung des Ankunftsstroms festgelegt werden. Dieses ermöglicht unter anderem die Modellierung von Parzellen, in denen vorzugsweise bestimmte Medientypen nachgefragt werden, wie es zum Beispiel bei einem klassischen Fernsehzimmer der Fall ist. Dieses ist notwendig, da häufige Benutzer des Systems das Vorhandensein bestimmter Ausgabegeräte in bestimmten Parzellen beachten werden, um zu verhindern, extra andere Parzellen aufsuchen zu müssen, um zum aktuellen Zeitpunkt ihre Dokumente zu betrachten, wenn sie sich ohnehin später in der Nähe eines entsprechenden Ausgabegerätes aufhalten. Die Verteilung der Marken wird als zeitunabhängig angenommen. Dieses ist für ein Büroszenarium unkritisch. Es muß jedoch betont werden, daß sich diese Annahme für Wohnumgebungen als falsch herausstellen könnte, da dort zum Beispiel abends im Wohnzimmer mit einer höheren Wahrscheinlichkeit Anfragen gestellt werden könnten als im Arbeitszimmer, es jedoch am vormittag umgekehrt sein könnte. Die genaue Anzahl der endlich vielen verschiedenen Marken ergibt sich aus der Anzahl der verschiedenen vorkommenden Medientypen multipliziert mit der Anzahl der Parzellen zu $7k$. Die diskrete Verteilung der Marken ist ebenfalls szenarienabhängig und wird daher jeweils bei der Beschreibung des Szenariums angegeben.

Um diese Marken zuzuordnen, erhält jeder Zeitpunkt S_n in einem weder von S_n noch von vorherigen Zuordnungen abhängigen zufälligen Experiment mit der Wahrscheinlichkeit $q_{(i,j)}$ die Marke $(i,j) \in \mathbb{X}$ mit

$$q_{(i,j)} \in [0,1], \ \sum_{i,j} q_{(i,j)} = 1, \tag{4.5}$$

d. h. es gilt

$$P(X_n = (i,j)) = q_{(i,j)}, \quad (i,j) \in \mathbb{X}, \quad n \in \mathbb{N}. \tag{4.6}$$

Den so modellierten allgemeinen Ankunftsstrom der Anfragen $\Phi = ((S_n, X_n))_{n \geq 1}$ kann man nun anhand der Marken X_n mit Hilfe von Indikatorfunktionen aufteilen. Dabei erhält man die Zählprozesse

$$A_{(i,j)}(t) = \sum_{n=1}^{\infty} \mathbf{1}_{\{S_n \leq t, X_n = (i,j)\}}, \quad (i,j) \in \mathbb{X}, \tag{4.7}$$

die die Anfragen getrennt nach Marken (i,j) angeben.

Satz 4.1. *Die $A_{(i,j)}(t)$, $(i,j) \in \mathbb{X}$, sind unabhängige Poisson-Prozesse mit den Intensitätsfunktionen $q_{(i,j)}\lambda(t)$.*

Beweis. Die hier betrachtete Aufteilung nach Marken ist ein Fall des sogenannten Bernoulli Samplings eines Poisson-Prozesses. Der Beweis hierzu ist in der Literatur [20] zu finden. □

Wir definieren

$$\lambda_{(i,j)}(t) := q_{(i,j)}\lambda(t), \quad (i,j) \in \mathbb{X} \tag{4.8}$$

um von den Ankunftsströmen $A_{(i,j)}$ mit ihren korrespondierenden Intensitätsfunktionen $\lambda_{(i,j)}(t)$ zu sprechen.

Handelt es sich bei den Ankunftszeitpunkten von Φ um einen inhomogenen Poisson-Prozeß, so sind auch die $A_{(i,j)}$ inhomogen, andernfalls bei $\lambda(t) \equiv \lambda$ ebenfalls homogen. Auch wenn man somit den multivariaten Ankunftsstrom Φ nach Marken (i,j) sortiert als einzelne inhomogene Poisson-Prozesse $\Phi_{(i,j)}$ mit Intensitätsfunktionen $\lambda_{(i,j)}(t)$ betrachtet, so bilden sie nach der Konstruktion zusammen einen inhomogenen Poisson-Prozeß mit der Intensitätsfunktion

$$\lambda(t) = \sum_{i,j} \lambda_{(i,j)}(t). \tag{4.9}$$

Der letzte zu betrachtende Parameter der Anfragen S_n ist die Darstellungslänge ihrer Multimedia-Daten Y_n. Diese Länge entspricht in der klassischen Terminologie der Warteschlangennetzwerke der Bedienzeit. Sie ist allein vom Medientyp abhängig und wird somit unabhängig von allen anderen Parametern wie Anfragezeitpunkt und -parzelle und Ausgabegerät bestimmt. Daher wird sie im folgenden Abschnitt gesondert betrachtet.

4.3.3 Darstellungslänge

Die Darstellungslänge wird in allen Szenarien als ausschließlich vom Medientyp abhängig angenommen. Da hierzu keine realen Daten vorliegen, muß auch hier mit Verteilungsannahmen gearbeitet werden. Eine Analyse des Benutzerverhaltens bei verschiedenen Medientypen zur Bestimmung der Verteilungen der Darstellungslängen wäre auch in einer herkömmlichen Büroumgebung möglich. Unter der Annahme, daß sich die Ergebnisse auf eine Ubiquitous Computing Umgebung übertragen ließen, könnte man in diesem Fall auf reale Daten zurückgreifen. Besonders interessant wäre hierbei die Überprüfung der Vermutung, daß die Darstellungslängen heavy-tailed verteilt sind. Crovella, Taqqu und Bestavros haben in ihrem Artikel „Heavy-Tailed Probability Distributions in the World Wide Web“ [2] unter anderem Dateigrößen von Audio- und Video-Dateien untersucht und festgestellt, daß diese heavy-tailed verteilt sind. Da bei Medientypen mit begrenzter zeitlicher Länge die Dateigröße mit der Länge korreliert sein sollte, ist dies ein Hinweis, der die oben angegebene Vermutung unterstützt.Wenn man z. B. eine Pareto-Verteilung, die die Dichte

$$f(x) = \alpha\beta^\alpha x^{-\alpha-1}, \quad \alpha, \beta > 0, \quad x \geq \beta \tag{4.10}$$

besitzt und als einfachste heavy-tailed Verteilung gilt, annimmt, könnte man versuchen, den Parameter α anhand der empirisch erhaltenen Daten zu schätzen. β dient zur Skalierung. Der Zusammenhang zwischen Dateigröße und Darstellungslänge des Medienstroms ist jedoch aufgrund verschiedener technischer Formate mit unterschiedlichen Datenkompressionsraten und verschiedenen Qualitätsstufen hinsichtlich der Auflösung selbst bei Videoströmen schwer zu evaluieren. Da keinerlei Untersuchungen über die Länge von Medienströmen vorliegen und das oben beschriebene Verfahren aufgrund der notwendigen Datenaquisition den Rahmen dieser Arbeit sprengen würde, muß darauf verzichtet werden.

In den später beschriebenen Simulationen werden daher die Darstellungslängen Y_n ausnahmslos als exponentialverteilt nach Definition 4.1 modelliert. Hierbei wird für jeden Medientyp i der Tabelle 2.1 ein spezieller Parameter μ_i bestimmt. Da der Erwartungswert der Exponentialverteilung $Exp(\lambda)$ durch $\frac{1}{\lambda}$ gegeben ist, lassen sich die μ_i als Reziproke der angenommenen durchschnittlichen Betrachtungsdauer des Medientyps i wählen. μ_i ist somit die Bedienrate des i-ten Medientyps.

4.3.4 Ankunftsstrom

Zur zusammenfassenden Beschreibung des Ankunftsstroms definieren wir zunächst

$$\mathbb{Y} := \mathbb{X} \times \mathbb{R}^+. \tag{4.11}$$

$\mathbb{Y}$ ist somit der Markenraum $\mathbb{X}$ erweitert um $\mathbb{R}^+$ zur Beschreibung der Darstellungslängen Y_n. $\mathcal{Y}$ bezeichne die Borelsche σ-Algebra in $\mathbb{Y}$. Der Ankunftsstrom liegt damit in der Form

$$\Psi = ((S_n, (X_n, Y_n)))_{n \geq 1}, \quad S_n \in \mathbb{R}^+, \quad (X_n, Y_n) \in \mathbb{Y} \tag{4.12}$$

vor. Hierbei ist zu beachten, daß X_n und Y_n keine unabhängigen Zufallsvariablen sind, da in X_n der Medientyp enthalten ist, der die Verteilung von Y_n bestimmt. Ψ ist ein markierter Poisson-Prozeß[1]. Dieser Name begründet sich auf

$$P(\Psi(A) = j) = \frac{\Lambda(A)^j}{j!} e^{-\Lambda(A)}, \quad j \in \mathbb{Z}^+, A \in \mathcal{B}^+ \otimes \mathcal{Y}, \tag{4.13}$$

mit dem Intensitätsmaß

$$\Lambda(B \times \{(i,j)\} \times C) := \left(\int_B \lambda(s) ds \right) q_{(i,j)} F_i(C). \tag{4.14}$$

Analog zu Definition 4.3 sind hierbei B und C Borelsche Teilmengen von $\mathbb{R}^+$ und $(i,j) \in \mathbb{X}$. F_i sei die vom Medientyp i aus der Marke (i,j) abhängige Verteilungsfunktion der Darstellungslängen. Die Meßbarkeit auf $\mathbb{R}^+ \times \mathbb{Y}$ bezieht sich auf die Produkt-σ-Algebra $\mathcal{B}^+ \otimes \mathcal{Y}$.

4.4 Leistungskenngrößen

Nachdem bisher der generelle Aufbau des Modells beschrieben wurde, sollen nun Qualitätsmaße bestimmt werden, mit deren Hilfe die Güte der Verteilung der Medienströme in einem gegebenen Szenarium beurteilt werden kann. Die beiden wichtigsten Kriterien sind einerseits die Wartezeit einer Anfrage, die bis zum Beginn ihrer Darstellung vergeht, und andererseits die Entfernung der Ausgabeparzelle von der Anfrageparzelle. Dieses stellt sich als ein Optimierungsproblem dar, da im Allgemeinen Anfragen desto schneller dargestellt werden können, je größere Entfernungen des darstellenden Gerätes vom Anfrageort in Kauf genommen werden. Das Verhalten des Verteilungsalgorithmus in dieser Hinsicht kann durch die in Abschnitt 3.2.2.1 besprochenen Parameter „Entfernungszeit“, „Trägheitsschwelle“ und „Ungeduldsschwelle“ gesteuert werden.

Sei W_i die Wartezeit, die die i-te Anfrage von ihrer Ankunft im System bis zum Beginn der Darstellung auf einem Ausgabegerät in der Warteschlange verbringt. Dann

[1] Mehr zur Theorie markierter Poisson-Prozesse und des folgenden Resultates sind in [12] zu finden.

erhält man bei n simulierten Anfragen die durchschnittliche beobachtete Wartezeit $\hat{E}(W)$ aller Anfragen anhand von

$$\hat{E}(W) = \frac{1}{n}\sum_{i=1}^{n} W_i, \tag{4.15}$$

sowie deren empirische Varianz

$$\hat{V}(W) = \frac{1}{n-1}\sum_{i=1}^{n}(W_i - \hat{E}(W))^2. \tag{4.16}$$

Hierbei wird durch $n-1$ und nicht durch n geteilt, um erwartungstreu zu schätzen. Analog lassen sich die durchschnittliche Wartezeit der Anfragen eines bestimmten Medientyps, einer Klasse, oder die Wartezeit, der in einer bestimmten Parzelle angeforderten Anfragen, untersuchen. Anhand dieser Werte lassen sich Schwachstellen im System, wie z. B. eine mangelnde Anzahl Ausgabegeräte für einen bestimmten Medientyp in einer Parzelle, finden.

Ferner sollte die mit der Wartezeit korrelierte Warteschlangenlänge betrachtet werden. Mit Hilfe der Indikatorfunktion

$$I_i(t) = \begin{cases} 1 & \text{falls} \quad S_i \le t < S_i + W_i, \\ 0 & \text{sonst,} \end{cases} \tag{4.17}$$

die 1 liefert, wenn sich die i-te Anfrage zur Zeit t in der Warteschlange befindet, errechnet sich die beobachtete Warteschlangenlänge zu

$$\hat{Q}(t) = \sum_{i=1}^{n} I_i(t). \tag{4.18}$$

Die Warteschlangenlänge sollte ebenfalls nicht nur allgemein sondern auch speziell für bestimmte Anfragetypen untersucht werden, wie bereits für die durchschnittliche Wartezeit angesprochen wurde.

Neben diesen Mittelwerten werden im konkreten Fall auch die Minima und Maxima, sowie die Varianzen der entsprechenden Größen betrachten. Die in diesem Abschnitt vorgestellten empirischen Erwartungswerte $\hat{E}$ und Varianzen $\hat{V}$ können bei einfachen Szenarien mit den induktiv berechenbaren Werte E und V verglichen werden. Es wird darauf dann im jeweiligen Fall im Kapitel 5 näher eingegangen.

4.5 Analyse

Wie in Abschnitt 4.1 begründet ist, muß auf eine Simulation zur Analyse des Gesamtsystems zurückgegriffen werden. Da die Simulationsergebnisse nicht mit empirischen Daten abgeglichen werden können, muß auf andere Weise die Korrektheit der Simulation validiert werden. Dieses ist zumindest gegenüber dem Modell möglich. Eine theoretische Untersuchung des gesamten Modells hinsichtlich der Güte der Verteilung der Medienströme verbietet sich aufgrund der Komplexität. In einfachen Spezialfällen lassen sich jedoch bestimmte Leistungskenngrößen berechnen oder mit Werten bekannter Warteschlangennetze aus der Literatur vergleichen. Da aufgrund der Interpretationsnotwendigkeit der Ergebnisse eine klare Trennung zwischen deduktiver mathematischer Vorgehensweise und Simulation nicht möglich ist, wird im nächsten Kapitel zunächst auf die Umsetzung des Modells in eine Simulation eingegangen. Im Abschnitt 5.3 werden dann Kenngrößen einfacher Szenarien mathematisch und simulationstechnisch bestimmt. Die Ergebnisse werden miteinander verglichen. Die Übereinstimmung der Werte ist als starker Hinweis auf die korrekte Umsetzung des Modells in die Simulation zu werten.

5. Simulation

In diesem Kapitel wird das mathematische Modell des vorangehenden Kapitels in eine Simulation umgesetzt, um die Leistungsfähigkeit des Verteilungssystems zu untersuchen. Zunächst wird dazu eine Einführung in das verwendete Simulationsprogramm und das zugrunde liegende Schema von Diskreten-Ereignis-Simulationen gegeben. Danach wird auf die Implementierung des in dieser Arbeit vorliegenden Modells eingegangen. Simuliert werden zunächst einfache Szenarien mit wenigen Ausgabegeräten, um die Korrektheit der Simulation zu validieren und spezielle Parameterabhängigkeiten zu untersuchen. Anschließend wird die Simulation der Büroumgebung des TecOs behandelt.

5.1 Diskrete-Ereignis-Simulation: OMNeT++

OMNeT++ steht für Objective Modular Network Testbed in C++. Es ist ein Werkzeug zur Simulation von Diskreten-Ereignis-Systemen (vgl. Abschnitt 5.1.1). Es erlaubt die Modellierung und Simulation von Computernetzwerken, Multiprozessor- und anderen verteilten Systemen. Es wurde zum Großteil von András Varga entwickelt, besitzt einen offenen Quellkode und ist kostenlos erhältlich [18]. Es ist in C++ geschrieben mit einer optionalen graphischen Oberfläche in Tcl/Tk, damit portierbar und es existieren Distributionen für vielerlei Entwicklungsumgebungen. Bei dieser Arbeit wurde die Version 2.0 mit graphischer Oberfläche unter Linux 2.2.10 mit X Window eingesetzt.

5.1.1 Diskrete-Ereignis-Simulation

In einem Diskreten-Ereignis-System geschehen Veränderungen des Zustandes des Gesamtsystems nur zu diskreten Zeitpunkten im Gegensatz zu kontinuierlichen Zustandsänderungen, wie sie z. B. durch Differentialgleichungen beschrieben werden. Diese Veränderungen werden Ereignisse genannt, sie benötigen keine Zeit um in Effekt zu treten. Zwischen Ereignissen bleibt der Systemzustand unverändert. Diskrete-Ereignis-Systeme können mit Hilfe der Diskreten-Ereignis-Simulation (DES) modelliert werden. Gleichzeitig eintretende Ereignisse werden parallel behandelt, es liegt am Modellierer hierbei auftretende Konflikte zu vermeiden. DES Modelle bestehen im allgemeinen aus stochastischen Komponenten, die die zufallsbedingte Natur des betrachteten Systems widerspiegeln. Simulationsprogramme arbeiten mit einem Zeitmechanismus, der jedem Ereignis einen Zeitstempel zuordnet. Die Simulationszeit ist unabhängig von der Real- oder CPU-Zeit. DES arbeiten mit einer Menge von zukünftigen Ereignissen, die im sogenannten Future Event Set (FES) abgelegt sind.

Algorithmus 5.1 DES-Schema

```
Initialisierung
while (FES nicht leer) and (Simulation nicht beendet) do
   entnehme nächstes Ereignis aus der FES
   Simulationszeit := Zeitstempel des Ereignisses
   bearbeite Ereignis
end while
beende DES
```

Algorithmus 5.1 zeigt schematisch die Arbeitsweise des Simulationsprogramms [19]. Bei der Initialisierung werden die benötigten Datenstrukturen angelegt und Ereignisse in der FES abgelegt, um die Simulation zu starten. In der folgenden Schleife werden Ereignisse der FES in der Reihenfolge ihrer Zeitstempel entnommen und bearbeitet. Die Bearbeitung geschieht durch benutzerdefinierten Programmkode und kann die Hinzufügung oder Entfernung von Ereignissen aus der FES beinhalten. Die Simulation wird beendet, wenn die FES leer ist oder wenn die Simulation ein Limit (Simulationszeit, CPU-Zeit, Statistikgenauigkeit, etc.) erreicht hat. Beim Beenden werden die gesammelten statistischen Daten ausgewertet.

5.1.2 Struktur

Ein OMNeT++ Modell besteht aus hierarchisch geschachtelten Modulen. Es stehen zwei Modultypen zur Verfügung.

Simple Modules bilden die unterste Hierarchiestufe und beinhalten vom Benutzer implementierten Kode, der die Bearbeitung der Ereignisse beschreibt und bei der Simulation ausgeführt wird.

Compound Modules setzen sich aus Simple Modules oder Compound Modules zusammen und enthalten keinen Kode. Sie dienen allein zur Beschreibung der Topologie der Simulation.

Module kommunizieren mit Hilfe von Nachrichten, die über gerichtete Kanten zwischen den Modulen ausgetauscht werden. Die Schnittstellen zwischen Modulen und den Verbindungen sind die sogenannten Gates. Nachrichten können beliebige Datenstrukturen tragen. Das Eintreffen einer Nachricht in einem Simple Module ist ein Ereignis des Diskreten-Ereignis-Systems. Dieses Ereignis wird bearbeitet, indem das

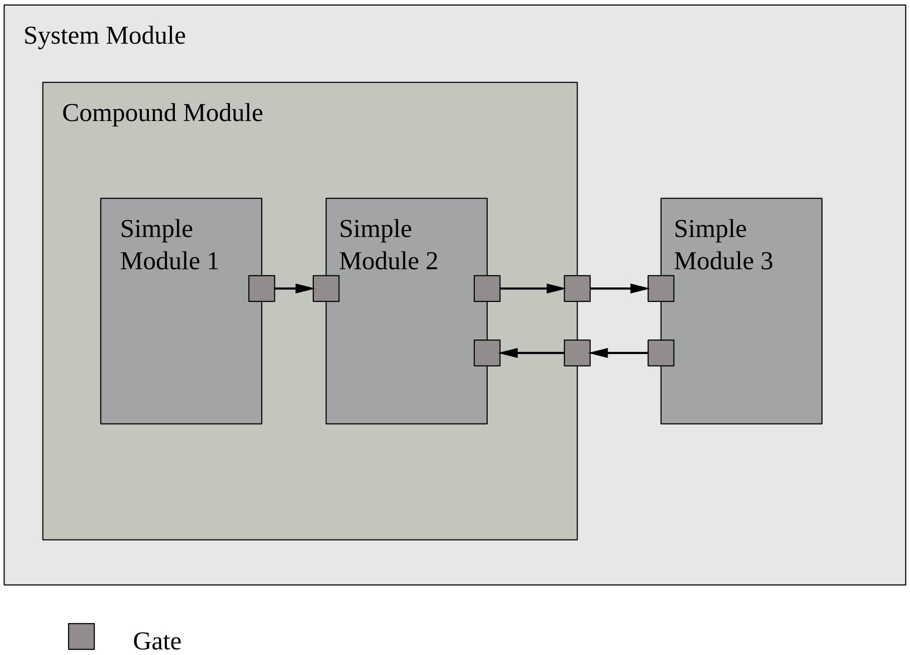

Abbildung 5.1: Topologie einer DES unter OMNeT++

Simple Module die Nachricht entsprechend seines Kodes verarbeitet. Abbildung 5.1 zeigt exemplarisch die Topologie einer DES unter OMNeT++. Die gesamte Simulation wird dabei ebenfalls durch ein Compound Module, dem System Module, repräsentiert.

5.1.3 Programmierung

Die Topologie des Modells wird in der OMNeT++ eigenen Network Description Language (NED) beschrieben. Hier werden für alle Module externe Parameter, Submodule, Gates und Verbindungen beschrieben. Verschiedene Verbindungstypen können ebenfalls anhand von Parametern modelliert werden. Als Beispiel nehme man den im Anhang A zu findenden Kode. Zur Erzeugung der Topologien steht zusätzlich ein graphischer Editor zur Verfügung.

Im Gegensatz zu Compound Modules, die nur durch ihre NED-Beschreibung definiert sind, besitzen Simple Modules zusätzlich eine Implementierung in C++, die die Nachrichtenverarbeitung regelt. Hierfür steht eine spezielle Simulator-Klassenbibliothek zur Verfügung. Im Anhang B ist eine Implementierung zu finden. OMNeT++ erlaubt sowohl eine Thread/Coroutine-basierte Programmierung als auch eine auf der Basis von endlichen Zustandsautomaten. Bei dieser Arbeit wurde ausschließlich die erste Methode verwendet.

5.2 Implementierung der Simulation

Es wurde ein Programm entwickelt, welches durch unterschiedliche Konfigurationsdateien zur Simulation verschiedenster Szenarien geeignet ist. Sowohl die mathematischen Verteilungen der Anfragen und Darstellungslängen, als auch die Parzellenaufteilung, sowie Anzahl und Eigenschaften der Ausgabegeräte werden anhand der Konfigurationsdatei bestimmt, so daß vergleichende Simulationsläufe mit geringen Änderungen schnell möglich sind. Die Implementierung der Simulation des HermiUC-Systems orientiert sich an der Struktur des mathematischen Modells (vgl. Abbildung 4.1). Alle Knoten des Netzwerks wurden als Simple Moduls realisiert, das gesamte Netzwerk ist eine Instanz eines Compound Modules. Abbildung 5.2 zeigt einen Bildschirmausdruck der Simulation des Medientyp-Vergleichsszenariums (vgl. Abschnitt 5.3.2). Der Generator dient als Anfragenquelle, der Controller beinhaltet die Warteschlange und den Auswahlalgorithmus, und die Ausgabegeräte simulieren die Darstellung der Medienströme. Im oberen Fenster der Abbildung 5.2 ist das linke Ausgabegerät als momentan belegt zu erkennen, wogegen der rechte Bildschirm leer ist, das Ausgabegerät somit unbelegt. Die Medienanfragen werden durch Nachrichten simuliert, die zwischen den Knoten ausgetauscht und falls notwendig im Controller in der Warteschlange zwischengespeichert werden. Man kann die Simulation mit graphischer Oberfläche in beliebiger Schnelligkeit ablaufen lassen, so daß bei Bedarf die Belegung der Ausgabegeräte und der Austausch von Nachrichten am Bildschirm verfolgt werden kann. In Abbildung 5.2 ist die Beauftragung des rechten Ausgabegerätes vom Controller mit der 2. Anfrage zu erkennen. Der Punkt mit der Bezeichnung `request-2` wandert entlang der Verbindung vom `controller` zum Ausgabegerät `ausgabe[1]`.

Für jede Anfrage ergibt sich somit folgender Ablauf:

1. Der Generator erzeugt eine neue Anfrage mit Anfrageparzelle, Medientyp und Darstellungslänge.

2. Der Generator sendet die Anfrage an den Controller.

3. Der Controller nimmt die Anfrage in die Warteschlange auf.

4. Der Controller untersucht die Warteschlange und sendet die Anfrage an ein freies Ausgabegerät.

5. Das Ausgabegerät wartet im Belegtzustand entsprechend der Darstellungslänge.

6. Das Ausgabegerät sendet die Anfrage an den Controller zur Statistikauswertung zurück.

7. Der Controller speichert die Statistikdaten und löscht die Anfrage.

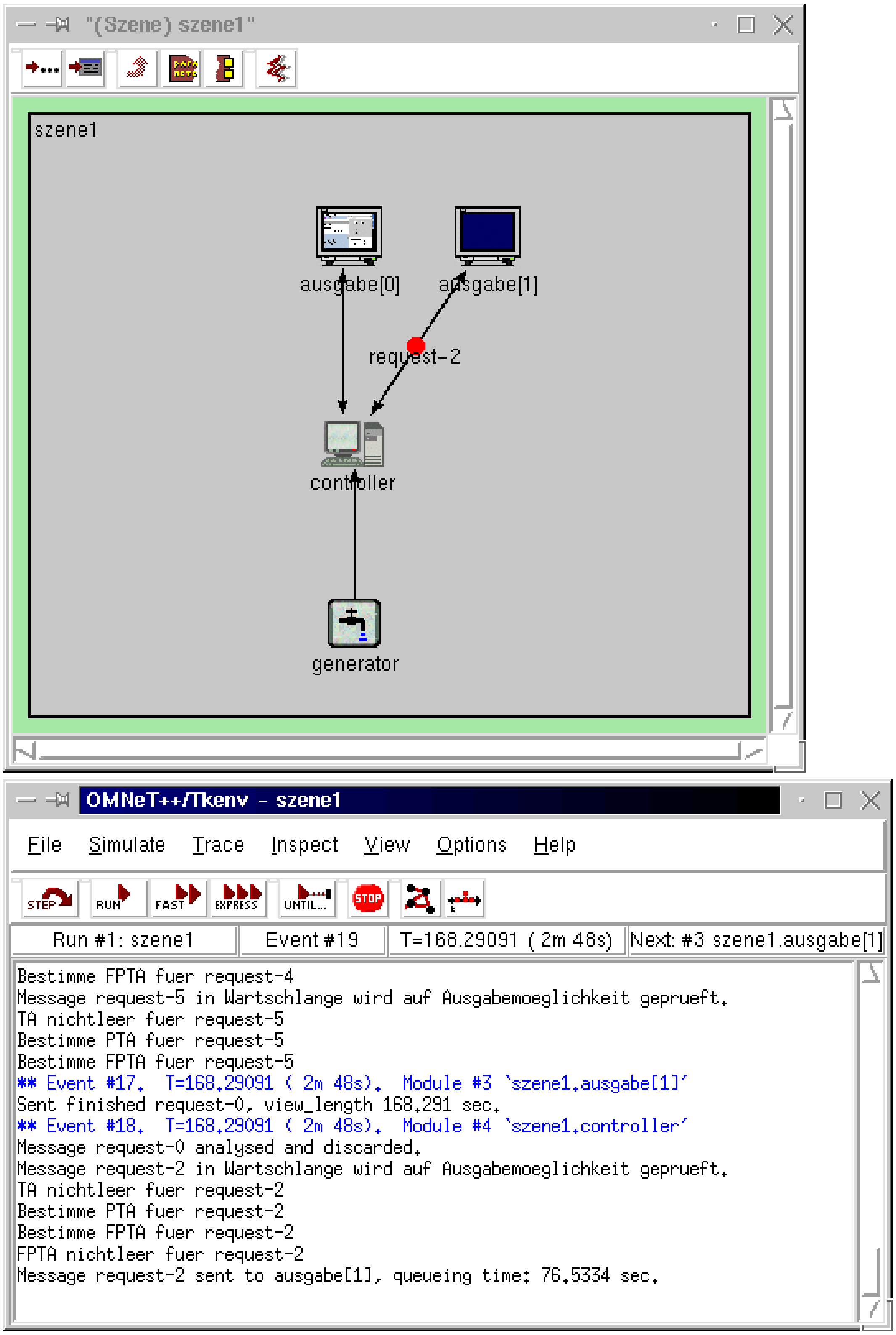

Abbildung 5.2: Bildschirmausdruck der Simulationsimplementierung

Neben der graphischen Darstellung lassen sich alle Daten der laufenden Simulation auch im Kontrollfenster (vgl. Abbildung 5.2 unten) verfolgen. Die Statistikdaten werden während der Simulation in Text-Dateien gespeichert, so daß sie anschließend zur Analyse mit belieben Statistikprogrammen zur Verfügung stehen. Die in Abschnitt 3.1.2 angesprochene Gerätedatenbank ist nicht explizit implementiert, da auf Parameter anderer Module innerhalb der Simulation durch spezielle Mechanismen zugegriffen werden kann. Der OMNeT++ NED Programmkode zur Beschreibung der Topologie ist in Anhang A abgedruckt. Beispielhaft ist der im Gegensatz zum Controller noch überschaubare C++ Programmkode des Generators in Anhang B ebenfalls zu finden. Die komplette Implementierung der Simulation umfaßt etwa zweitausend Kodezeilen.

5.3 Vergleichsszenarien

Bevor als realistische Umgebung die Simulation der Büroräumlichkeiten des TecOs beschrieben wird, werden zunächst, wie bereits in Abschnitt 4.5 angekündigt, Spezialfälle mathematisch und simulationstechnisch analysiert. Hierbei soll gezeigt werden, daß die Simulationsergebnisse mit den mathematischen Berechnungen übereinstimmen. Dadurch lassen sich Fehler bei der Umsetzung des mathematischen Modells in die Simulation weitestgehend ausschließen.

Bei den Simulationsläufen wird als Basiszeiteinheit die Sekunde verwendet, daher sind alle im folgenden gemachten Zeitangaben ebenfalls als solche zu interpretieren, sofern nicht explizit anderes genannt wird.

5.3.1 M/M/2-Warteschlangennetz

Zunächst wurde das in der Warteschlangentheorie schon vielfach untersuchte M/M/2-Warteschlangenetz simuliert. Die in der Literatur übliche von Kendall eingeführte Notation M/M/2 ist folgendermaßen zu interpretieren. Das erste M zeigt an, daß der Ankunftsprozeß gedächtnislos (memoryless) ist, d. h. die Zwischenankunftszeiten unabhängig identisch exponentialverteilt sind. Es handelt sich somit um einen Poisson-Prozeß der durch seine Ankunftsrate λ spezifiziert wird. Das zweite M bezieht sich auf gedächtnislos verteilte Bedienzeiten. Diese sind sowohl untereinander als auch vom Poisson-Prozeß unabhängig identisch $Exp(\mu)$ verteilt. Die 2 gibt die Anzahl der Server im System an. Es wird die FCFS-Methodik angewendet. Dieses Netzwerk tritt im HermiUC-System als Spezialfall auf, wenn sich zwei identische Ausgabegeräte, die nur einen Medientyp unterstützen, in einer Parzelle befinden, in der keine weiteren Ausgabegeräte vorhanden sind, die diesen Typ ebenfalls darstellen können. Es wurden vier Simulationsdurchläufe mit verschiedenen Startwerten für die verwendeten Zufallsgeneratoren gerechnet. Bei jedem Durchlauf wurden eine Millionen Anfragen simuliert. Es wurde $\lambda = \frac{1}{5}$ für die Ankunftsrate und $\mu = \frac{1}{3}$

für die Bedienrate verwendet. Die Stabilitätsbedingung für diese System ist durch $\frac{\lambda}{2\mu} < 1$ gegeben [20]. Daher repräsentiert die hier vorliegende Wahl der Raten mit $\frac{\lambda}{2\mu} = \frac{3}{10}$ eine leichte bis mittlere Belastung des Systems, bei der Simulationen im Allgemeinen stabilere Werte liefern als unter hoher Belastung.

Quelle	Mittlere Anzahl von Kunden	Mittlere Wartezeit
1. Durchlauf	0,661	0,299
2. Durchlauf	0,660	0,297
3. Durchlauf	0,660	0,296
4. Durchlauf	0,658	0,297
Mittelwerte der Durchläufe[1]	0,660	0,297
theoretischer Wert	0,659	0,297

Tabelle 5.1: Analyse des M/M/2-Warteschlangennetzes

Tabelle 5.1 zeigt die erhaltenen Größen. Bei der mittleren Anzahl von Kunden im System sind die sich momentan in Bedienung befindenden Kunden sowie die Anfragen in der Warteschlange eingeschlossen. Bei allen Simulationdurchläufen wird dieser Wert nicht kontinuierlich erfaßt, sondern immer dann notiert, wenn eine neue Anfrage erzeugt wird, jedoch bevor diese selbst in die Warteschlange aufgenommen wird. Da unter bestimmten Unabhängigkeitsbedingungen, die in den hier betrachteten Szenarien immer erfüllt sind, Ankunftszeiten eines homogenen Poisson-Prozesses Zeitmittelwerte von Größen sehen, ist dieses Vorgehen beim Schreiben der Statistikdaten legitim.[2] Die mittlere Wartezeit gibt die durchschnittliche Aufenthaltszeit von Anfragen in der Warteschlange ohne ihre folgende Bedienzeit an. Die theoretischen Werte ergeben sich aus

$$E(Q) = \frac{4\lambda\mu}{4\mu^2 - \lambda^2} \tag{5.1}$$

für die mittlere Anzahl von Kunden und

$$E(W) = \frac{E(Q)}{\lambda} - \frac{1}{\mu} \tag{5.2}$$

für die mittlere Wartezeit [20]. Gleichung 5.2 ist ein Spezialfall der in vielfachen Szenarien anwendbaren Little Formel, die meisten in der Form

$$E(Q) = \lambda E(A) \tag{5.3}$$

geschrieben wird, wobei $E(A)$ die mittlere Aufenthaltszeit von Anfragen im System, d. h. die Wartezeit in der Warteschlange inklusive der folgenden Bedienzeit,

[1] Alle Werte sind auf 3 Dezimalstellen gerundet, die Berechnung der Mittelwerte erfolgte jedoch hier und in den folgenden Tabellen immer mit 6 Dezimalstellen.

[2] Diese vielgenutzte Eigenschaft ist in der englischsprachigen Literatur unter PASTA (Poisson arrivals see time averages) zu finden. Die genauen Zusammenhänge und Bedingungen sind z. B. in [20] nachzulesen.

beschreibt. Bei dem Ausgleichsterm $\frac{1}{\mu}$ in Gleichung 5.2 handelt es sich um die mittlere Bediendauer der Anfragen. Dieser würde entfallen, wenn man anstatt der mittleren Gesamtanzahl von Anfragen im System $E(Q)$ nur die mittlere Anzahl von Anfragen in der Warteschlange verwendet, ohne die gerade sich in Bedienung Befindenden mitzuberücksichtigen. Die geringen Schwankungen der erfaßten Größen zwischen den Simulationsläufen und die Übereinstimmung mit den nach der Theorie zu erwartenden Werten weißt auf eine stabile und korrekte Simulation hin.

Im folgenden werden nun Szenarien betrachtet, bei denen eine direkte Überprüfung der Ergebnisse mittels Formeln wie Gleichung 5.1 nicht mehr möglich ist. Dieses liegt darin begründet, daß diese Bedienungssysteme offenbar in der Literatur bisher nicht behandelt wurden und eine mathematische Herleitung ähnlicher Formeln weder der Fokus dieser Arbeit ist noch aufgrund des Umfangs hier geleistet werden kann.[3]

5.3.2 Medientyp-Vergleichsszenarium

Nun wird ein Szenarium mit zwei Ausgabegeräten betrachtet, die sich in derselben Parzelle befinden (vgl. Abbildung 5.3). Daher muß die zweite Komponente, die die Anfrageparzelle bestimmt, der in Abschnitt 4.3.2 angesprochenen Marken hier nicht weiter berücksichtigt werden. Die Ausgabegeräte unterscheiden sich in der Menge der unterstützten Medientypen. Ausgabegerät 0 unterstützt nur den Medientyp 1, Ausgabegerät 1 dagegen kann Anfragen der Medientypen 1 und 2 darstellen. Dadurch soll die gegenseite Beeinflussung unterschiedlicher Medientypen, die teilweise um dasselbe Ausgabegerät konkurrieren, untersucht werden. Die Anfragezeitpunkte werden als homogener Poisson-Prozeß mit der Ankunftsrate $\lambda = \frac{1}{5}$ modelliert. Den Anfragen werden nach der Erzeugung durch ein Bernoulli-Experiment mit $q = \frac{1}{2}$ der Medientyp 1 oder 2 zugeordnet. Diese Zuordnung geschieht wie in Abschnitt 2.2.3 begründet unabhängig vom Poisson-Prozeß. Die Darstellungslängen sind für beide Medientypen identisch stochastisch unabhängig $Exp(\frac{1}{3})$ verteilt. Da nur eine Parzelle vorliegt, haben die im Auswahlalgorithmus vorkommenden Parameter Entfernungszeit, Trägheitsschwelle und Ungeduldsschwelle keinen Einfluß auf die Simulation.

Es ergibt sich damit folgendes Bedienverhalten im Netzwerk. Sofern ein geeignetes Ausgabegerät frei ist wird strikt eine FCFS-Methode unter allen Anfragen unabhängig ihres Medientyps angewendet. Wenn beide Ausgabegeräte frei sind, wird Medientyp 1 aufgrund der höheren Typspezialisierung immer vom Ausgabegerät 0 bedient. Die FCFS-Methodik wird nur dann übergangen, wenn eine Anfrage mit Medientyp 2 nicht dargestellt werden kann, da Ausgabegerät 1 belegt ist. In diesem Fall wird die älteste Anfrage des Medientyps 1 zum Ausgabegerät 0 geschickt, sofern dieses frei ist.

Es wurden vier Simulationsläufe mit je einer Millionen Anfragen gerechnet. Tabelle 5.2 zeigt die Simulationergebnisse. $\hat{E}(Q)$, bzw. $\hat{E}(Q^{(i)})$, ist die empirisch beobachtete mittlere Anzahl von Anfragen im System unabhängig ihres Typs, bzw. für

[3] Man vergleiche hierzu auch die Bemerkung am Ende des nächsten Abschnitts.

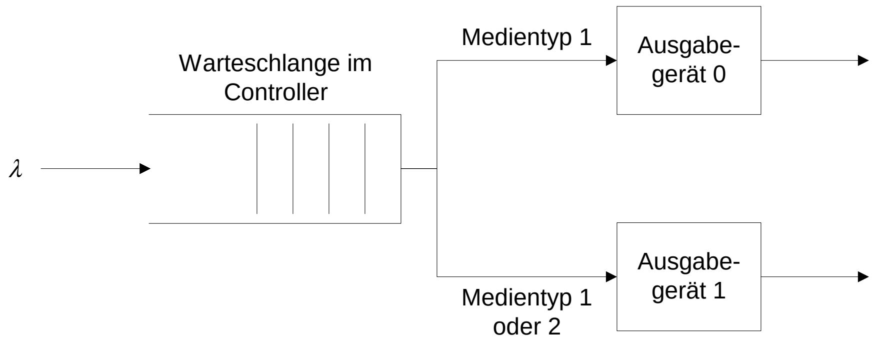

Abbildung 5.3: Struktur des Medientyp-Vergleichsszenariums

Quelle	$\hat{E}(Q)$	$\hat{E}(W)$	$\hat{E}(W^{(1)})$	$\hat{E}(W^{(2)})$
1. Durchlauf	0,779	0,892	0,227	1,556
2. Durchlauf	0,780	0,895	0,234	1,557
3. Durchlauf	0,779	0,900	0,228	1,568
4. Durchlauf	0,777	0,889	0,229	1,549
Mittelwert	0,779	0,894	0,230	1,557

Tabelle 5.2: Simulationsergebnisse des Medientyp-Vergleichsszenariums mit gleichen Raten

Anfragen des Typs i, $\hat{E}(W)$, bzw. $\hat{E}(W^{(i)})$, bezeichnet die beobachtete mittlere Aufenthaltszeit in der Warteschlange für alle Anfragen, bzw. für Anfragen des Typs i $(i = 1, 2)$.

Mit Hilfe der Little Formel läßt sich die mittlere Anzahl der Anfragen im System getrennt nach Typen berechnen. Aus Gleichung 5.2 erhält man

$$E(Q^{(i)}) = \lambda_i E(W^{(i)}) + \frac{\lambda_i}{\mu_i}, \quad i = 1, 2. \tag{5.4}$$

Hierbei ist in diesem Fall $\mu_i = \mu = \frac{1}{3}$ und $\lambda_i = \frac{\lambda}{2} = \frac{1}{10}$ $(i = 1, 2)$. Mit $\hat{E}(W^{(i)})$ $(i = 1, 2)$ aus Tabelle 5.2 ergibt sich $\hat{E}(Q^{(1)}) = 0.323$ und $\hat{E}(Q^{(2)}) = 0.456$. Man beachte zur Überprüfung, daß die Gleichung

$$\hat{E}(Q) = \hat{E}(Q^{(1)}) + \hat{E}(Q^{(2)}) \tag{5.5}$$

erfüllt ist. Wichtig ist dabei anzumerken, daß die Anwendung von Formeln wie Gleichung 5.4 nur dann korrekt ist, wenn sich daß jeweilige System im stationären Regime befindet. Dieses wurde mittels genügend großer Simulationsdurchläufe, verschiedenen Analysen der empirisch erhaltenen Daten wie z. B. Beobachtung der zeitabhängigen Mittelwerte auf $[0, t]$ und Wahl von geringen bis mittleren Belastungen der Systeme sichergestellt.

Die mittlere Wartezeit und Anzahl der Anfragen liegt höher als beim vorher betrachteten M/M/2-Bedienungssystems. Die Werte des M/M/2-Netzwerkes bilden offenbar eine Untergrenze für alle anderen Bedienmethoden in einem Netzwerktyp mit zwei Servern. Dies liegt darin begründet, daß beim oben betrachteten M/M/2-System niemals Anfragen in der Warteschlange verbleiben, obwohl ein Server frei ist, so daß ein Minimum an Wartezeit erreicht wird. Da im hier betrachteten Szenarium in bestimmten Fällen Anfragen in der Warteschlange verbleiben, obwohl ein Server frei ist, ist das System vermutlich nicht mehr stabil für alle $\frac{\lambda_1+\lambda_2}{\mu_1+\mu_2} < 1$. Die Wartezeit und nach der Little Formel damit auch die Warteschlangenlänge von Anfragen des Typs 2 ist signifikant länger als die des Typs 1. Man könnte die Differenz der Wartezeiten zwischen den Typen und auch die typunabhängige Wartezeit $E(W)$ verringern, wenn man Typ 2 Priorität bei der Bedienung auf Ausgabegerät 1 einräumte [7]. Wie in Abschnitt 3.2.2.1 begründet trifft dieses Vorgehen jedoch bei Menschen auf Unverständnis in vielen Situationen. Daher wird davon abgesehen.

Bei den Büroszenarien (Abschnitt 5.4) werden wir von unterschiedlichen Ankunftsraten und Bedienraten zwischen den Medientypen ausgehen. Daher soll auch dieser Fall hier untersucht werden. Betrachten wir dazu nun ein Szenarium, in dem zwei verschiedene Typen als Poisson-Prozesse mit den Ankunftsraten $\lambda_1 = \frac{3}{20}$ und $\lambda_2 = \frac{1}{20}$ eintreffen. Die zusammengenommene Ankunftsrate ist mit dem vorherigen Szenarium identisch: $\lambda = \lambda_1 + \lambda_2 = \frac{1}{5}$. Die Bedienzeiten sind typabhängig $\mu_1 = \frac{1}{3}$ und $\mu_2 = \frac{1}{2}$ unabhängig voneinander und von den Poisson-Prozessen exponentialverteilt. Die Netzwerktopologie inklusive der Medientypspezialisierung der Ausgabegeräte wird beibehalten. Es wurden erneut vier Durchläufe mit je einer Millionen Anfragen simuliert. Tabelle 5.3 zeigt die dabei erhaltenen Mittelwerte.

$\hat{E}(Q)$	$\hat{E}(W)$	$\hat{E}(W^{(1)})$	$\hat{E}(W^{(2)})$
0,617	0,335	0,219	0,683

Tabelle 5.3: Simulationsergebnisse des Medientyp-Vergleichsszenariums mit unterschiedlichen Raten

Mit Hilfe der Gleichungen 5.4 und 5.5 lassen sich erneut die Warteschlangenverteilungen untersuchen. Bei den vorherigen Szenarien, bei denen die Ankunftsraten aller nach Typen getrennten Poisson-Prozesse identisch waren, war $E(W)$ jeweils der Mittelwert der $E(W^{(i)})$. Bei unterschiedlichen Ankunftsraten muß jedoch die Gewichtung mitberücksichtigt werden. Mittels

$$E(W) = q_1 E(W^{(1)}) + q_2 E(W^{(2)}) \tag{5.6}$$

kann im allgemeinen Fall die mittlere Wartezeit einer beliebigen Anfrage unabhängig von ihrem Typ berechnet werden. Dabei ist

$$q_i = \frac{\lambda_i}{\lambda}, \quad i = 1, 2, \tag{5.7}$$

d. h. q_i gibt an, mit welcher Wahrscheinlichkeit eine Anfrage des Gesamtstroms vom Typ i ist. Dieses ist äquivalent zu der Modellierung eines Gesamtankunftsstroms und anschließender Typzuweisung nach den Wahrscheinlichkeiten q_i.

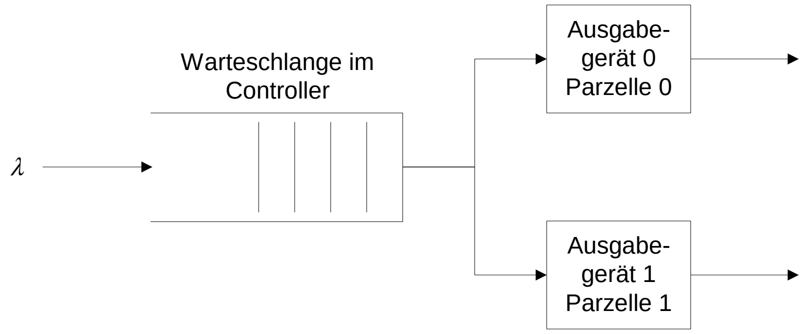

Abbildung 5.4: Struktur des Parzellenwechsel-Vergleichsszenariums

Ein ähnliches Bedienungssystem jedoch mit Prioritäten ist in [7] mathematisch untersucht worden. Dort läßt sich der Umfang solcher Analysen erkennen, der zu eigenstehenden Arbeiten berechtigt und hier nicht durchgeführt werden kann. Der größte Nachteil besteht bei einer rein simulationstechnischen Untersuchung darin, daß auf exakte Aussagen über Stabilitätsbedingungen verzichtet werden muß.

5.3.3 Parzellenwechsel-Vergleichsszenarium

In diesem Szenarium soll die Verteilung von Anfragen über Parzellengrenzen hinweg untersucht werden. Es liegen weiterhin nur zwei Ausgabegeräte vor, von denen sich jedoch Ausgabegerät i in Parzelle i ($i = 0, 1$) befindet. Weitere leere Parzellen sind nicht vorhanden. Bei den Medientypen beschränken wir uns hier auf einen Typ, den beide Geräte gleichermaßen darstellen können. Um Vergleiche mit den vorherigen Szenarien zu ermöglichen sind die Darstellungslängen weiterhin $Exp(\frac{1}{3})$ verteilt, der homogene Poisson-Prozeß der Anfragezeitpunkte besitzt erneut die Ankunftsrate $\lambda = \frac{1}{5}$. Die Anfrageparzellen werden analog zu den Medientypen im Medientyp-Vergleichsszenarium durch ein Bernoulli-Experiment mit $q = \frac{1}{2}$ bestimmt. Abbildung 5.4 zeigt den Aufbau des Netzwerkes. Der Parameter Ungeduldsschwelle des Auswahlalgorithmus ist auf 4 Sekunden gesetzt. Die anderen Parameter des Algorithmus 3.1 sind so gewählt, daß die Bedingung „geschätzte Wartezeit auf AG in Anfrageparzelle - Entfernungszeit $>$ Trägheitsschwelle“ nie erfüllt ist. Sie spielen daher bei der Simulation keine Rolle.

Mit diesen Einstellungen ergibt sich folgendes Verhalten. Anfragen werden in der Anfrageparzelle dargestellt, sobald das Ausgabegerät dort frei ist. Ist dies nach 4 Sekunden in der Warteschlange nicht erfüllt, wird die Anfrage auf dem nächsten freien Ausgabegerät unabhängig von der Parzelle dargestellt. Sollte also beim Überschreiten der Ungeduldsschwelle das Ausgabegerät in der anderen Parzelle frei sein, wird die Ausgabe dort nach exakt 4 Sekunden Aufenthaltszeit in der Warteschlange gestartet. Vor Ablauf der Ungeduldsschwelle findet niemals eine Bedienung in der Nicht-Anfrageparzelle statt, auch wenn das dortige Ausgabegerät bereits frei ist.

Quelle	$\hat{E}(Q)$	$\hat{E}(W)$	$\hat{P}$(Parzellenwechsel)
1. Durchlauf	0,770	0,839	0,079
2. Durchlauf	0,768	0,839	0,080
3. Durchlauf	0,766	0,834	0,079
4. Durchlauf	0,768	0,840	0,079
Mittelwert	0,768	0,838	0,079

Tabelle 5.4: Simulationsergebnisse des Parzellenwechsel-Vergleichsszenariums

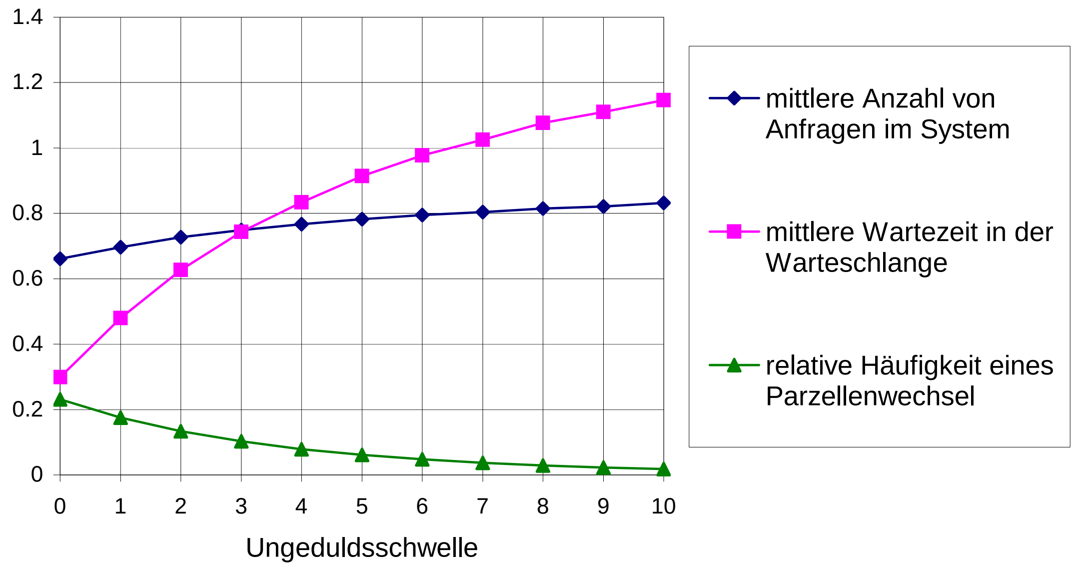

Abbildung 5.5: Einfluß der Ungeduldsschwelle im Parzellenwechsel-Vergleichsszenarium

Im ersten Versuch wurden vier Simulationsläufe mit je einer Millionen Anfragen und gleichbleibender Ungeduldsschwelle umgesetzt. Die Ergebnisse sind in Tabelle 5.4 zu sehen. $\hat{P}$(Parzellenwechsel) gibt die empirisch beobachtete relative Häufigkeit an, mit der eine Anfrage nicht in der Parzelle dargestellt wurde, in der sie angefordert wurde. D. h. sei $X_i = 0$, falls die i-te Anfrage in ihrer Anfrageparzelle dargestellt wird, und 1 sonst. Dann ergibt sich bei n simulierten Anfragen die relative Häufigkeit eines Parzellenwechsels zu

$$\hat{P}(\text{Parzellenwechsel}) = \frac{1}{n}\sum_{i=1}^{n} X_i. \tag{5.8}$$

Bei nur 8% umgelenkten Anfragen liegt die mittlere Wartezeit noch unter der des ersten betrachteten Szenariums in Abschnitt 5.3.2. Nun soll untersucht werden, wie stark die Ungeduldsschwelle die Kenngrößen beeinflußt. Dazu wurden 11 Simulationsdurchläufe gefahren, bei denen der Parameter der Ungeduldsschwelle von 0 bis 10 variiert wurde. Bei je einer Millionen simulierter Anfragen wurden Ergebnisse gemäß Abbildung 5.5 erhalten.

Hier zeigt sich, daß die mittlere Wartezeit und die relative Häufigkeit eines Parzellenwechsels relativ zu ihren Größen stark unter der Ungeduldsschwelle variieren. $\hat{E}(W)$ nimmt dabei über das gesamte hier betrachtete Intervall um 283% zu, $\hat{P}$(Parzellenwechsel) sinkt dagegen um 92%. Die Ungeduldsschwelle sollte nicht zu niedrig angesetzt werden, da ansonsten sehr viele der für die Benutzer unangenehmen Parzellenwechsel generiert werden. Daher sollte überlegt werden, welche mittlere Wartezeit noch zumutbar ist. Bei relativ zur mittleren Darstellungsdauer großen Werten für die Ungeduldsschwelle zeigt sich eine näherungsweise lineare Abhängigkeit. Da jedoch eine Wartezeit, die weit über die Darstellungsdauer der Anfrage hinausgeht, für den Anfragesteller nicht akzeptabel ist, sollte die optimale Einstellung des Parameters im nichtlinearen Bereich der in Abbildung 5.5 dargestellten Größen gesucht werden. Soweit es sich aus diesem kleinen Stichprobenraum folgern läßt, scheint auch hier die Little Formel erfüllt zu sein. In Abschnitt 5.4.4 wird darauf eingegangen in wieweit sich diese Einschätzungen auf größere Szenarien übertragen lassen.

5.4 Büroszenarien

Als realistisches Anwendungsszenarium wurde die Büroumgebung des Telecooperation Office (vgl. Abbildung 3.1) in eine Simulation umgesetzt. Der Erreichbarkeitsgraph wurde nach Abbildung 3.3 implementiert. Die Verteilung der Ausgabegeräte mit ihren unterstützten Medientypen auf die Parzellen zeigt Tabelle 5.5. Die simulierte Ausstattung der Büroumgebung orientiert sich an der real vorhandenen. Sie ist lediglich um einige Ausgabegeräte erweitert, die bisher aus Kostengründen nur in geringer Zahl oder noch gar nicht vorhanden sind, in wenigen Jahren jedoch zum Standard gehören könnten. Darunter fallen zum einen die als elektronische Tafeln bereits im Anwendungsszenarium (Abschnitt 1.3) vorgestellten Wandbildschirme, bei denen es sich um große Flachbildschirme handelt, die schon heute in Besprechungsräumen eingesetzt werden. Zum anderen handelt es sich um elektronische Bilder, die z. B. die Aufgabe von Wandkalendern oder Informationstafeln übernehmen. Diese sind mit Hilfe von elektronischem Papier realisiert und eignen sich daher nicht zu Bewegtbilddarstellung, weshalb ihnen nur der Medientyp 5 zugeordnet wird. Sie sind in größerer Zahl vorhanden, da sie überall dort sinnvoll eingesetzt werden können, wo sich heutzutage Poster oder andere Bilder befinden. Personal Computer mit Lautsprechern sind in der Tabelle als Audio Computer aufgeführt, Bildschirm bezeichnet einen Monitor ohne Audioausgabemöglichkeit, Audio Wandbildschirme sind mit Lautsprechern gekoppelte Wandbildschirme. Die Darstellungslängen der verschiedenen Medientypen i wurden als stochastisch unabhängig $Exp(\frac{1}{\mu_i})$ verteilt angenommen. Tabelle 5.6 zeigt die verwendeten Bedienraten.

Der inhomogene Poisson-Prozeß zur Bestimmung der Anfragezeitpunkte wurde wie folgt festgelegt. Zunächst wurde im Zuge dieser Diplomarbeit eine Umfrage über die Arbeitszeitverteilung und das Medienverhalten unter den Mitarbeitern des TecOs durchgeführt. Die Ergebnisse der Umfrage wurden mittels linearer Splines [17] in-

Pz.	Nr.	Gerät	Typ
0	0	Radio	1-2
	1	Audio Computer	1-7
	2	Wandbildschirm	5-7
1	3	Audio Computer	1-7
2	4	Audio Computer	1-7
	5	Bildschirm	5-7
	6	Radio	1-2
3	7	Fernseher	3-4
	8	Audio Computer	1-7
	9	elektron. Bild	5
	10	Radio	1-2
4	11	Analogradio	1
	12	elektron. Bild	5
	13	Audio Computer	1-7
	14	elektron. Bild	5
	15	Wandbildschirm	5-7
5	16	Radio	1-2
	17	Fernseher	3-4
	18	Audio Computer	1-7
	19	elektron. Bild	5
	20	Bildschirm	5-7
6	21	Radio	1-2
	22	Audio Computer	1-7
	23	Wandbildschirm	5-7
7	24	Audio Wandb.	1-7
	25	Wandbildschirm	5-7
	26	Radio	1-2
	27	elektron. Bild	5
	28	elektron. Bild	5
8	29	Audio Computer	1-7
	30	Audio Computer	1-7
	31	Audio Computer	1-7
	32	Audio Computer	1-7
	33	elektron. Bild	5
	34	elektron. Bild	5
	35	Bildschirm	5-7
	36	Bildschirm	5-7
	37	Bildschirm	5-7
9	38	Wandbildschirm	5-7
10	39	elektron. Bild	5
11	40	elektron. Bild	5
12	41	elektron. Bild	5
	42	Audio Computer	1-7
13	43	Audio Computer	1-7
	44	elektron. Bild	5
	45	Bildschirm	5-7
	46	Bildschirm	5-7
15	47	elektron. Bild	5
17	48	elektron. Bild	5
	49	Audio Wandb.	1-7
18	50	Bildschirm	5-7
	51	Bildschirm	5-7
	52	Audio Computer	1-7
	53	Audio Computer	1-7
	54	elektron. Bild	5
	55	Bildschirm	5-7
	56	Bildschirm	5-7
19	57	elektron. Bild	5
20	58	elektron. Bild	5
21	59	Audio Wandb.	1-7
22	60	Wandbildschirm	5-7
	61	elektron. Bild	5
23	62	Fernseher	3-4
	63	Bildschirm	5-7
	64	elektron. Bild	5
	65	Bildschirm	5-7
	66	Radio	1-2
24	67	elektron. Bild	5
	68	Audio Computer	1-7
	69	elektron. Bild	5
	70	Wandbildschirm	5-7
	71	Bildschirm	5-7
25	72	Radio	1-2
	73	Audio Computer	1-7
26	74	elektron. Bild	5
	75	Wandbildschirm	5-7
	76	Wandbildschirm	5-7
	77	Audio Computer	1-7
	78	Audio Computer	1-7
	79	elektron. Bild	5

Tabelle 5.5: Verteilung der Ausgabegeräte auf die Parzellen

Medientyp i	Beispiel	Bedienrate $\frac{1}{\mu_i}$
1	Radio	600
2	Tondokument	180
3	Fernsehen	600
4	Tonfilm	100
5	Bild/Text	30
6	Webcam	15
7	Stummfilm	60

Tabelle 5.6: Bedienraten der Medientypen

terpoliert. Ein 24h-Tag wurde dazu in regelmäßige Teilintervalle von 30 Minuten Länge eingeteilt. Die so erhaltene Intensitätsfunktion $\lambda(t)$ ist programmtechnisch folgendermaßen implementiert. Der Programmkode hierzu ist im Generator-Modul enthalten und im Anhang B abgedruckt. Zunächst wird durch exponentialverteilte Zwischenankunftszeiten ein homogener Poisson-Prozeß $\{S_n, n \geq 1\}$ mit der Ankunftsrate

$$\lambda := \max_t \lambda(t) \tag{5.9}$$

erzeugt. Anschließend werden die so erhaltenen Zeitpunkte durch ein Bernoulli-Experiment mittels der normierten Akzeptanzfunktion

$$p(t) := \frac{\lambda(t)}{\lambda} \tag{5.10}$$

weiterverarbeitet. Mit der Wahrscheinlichkeit $1 - p(S_n)$ werden sie verworfen und sind nicht Teil des inhomogenen Poisson-Prozesses [15]. Die Funktion $p(t)$ repräsentiert die zahlenmäßige Anwesenheit der Mitarbeiter des TecOs innerhalb der 24 Stunden eines normalen Arbeitstags normiert auf das Maximum 1. Sie ist in Abbildung 5.6 dargestellt. Bei der Simulation von mehreren Tagen werden die S_n des homogenen Poisson-Prozesses mit der Erfolgswahrscheinlichkeit $p(S_n \bmod 24h)$ in den inhomogenen Poisson-Prozeß aufgenommen.

Die Güte der Verteilung der Medienströme wurde bei Simulationsläufen mit unterschiedlichen Parametern untersucht. Hierbei wurden sowohl die drei Parameter des Auswahlalgorithmus variiert, als auch die Ankunftsrate λ und die Verteilung der Marken $q_{(i,j)}$. Bei der Verteilung der Marken wurde jedoch in allen Fällen davon ausgegangen, daß jede Anfrage mit gleicher Wahrscheinlichkeit aus jeder Parzelle stammt. Damit sollte einerseits eine unausgewogene Verteilung auf verschiedenen Parzellen verhindert werden. Andererseits würde eine Verteilung der Marken entsprechend den vorhandenen Ausgabegeräten in einer Parzelle dazu führen, daß alle Anfragen in ihrer jeweiligen Anfrageparzelle dargestellt werden. Ein solches gezieltes Anfragen der Medientypen ist nur dann zu erwarten, wenn sich die Benutzer perfekt mit der vorhandenen Infrastruktur auskennen. Wie jedoch schon in der Motivation (Abschnitt 1.2) erläutert, ist davon in der Zukunft aufgrund der steigenden Anzahl

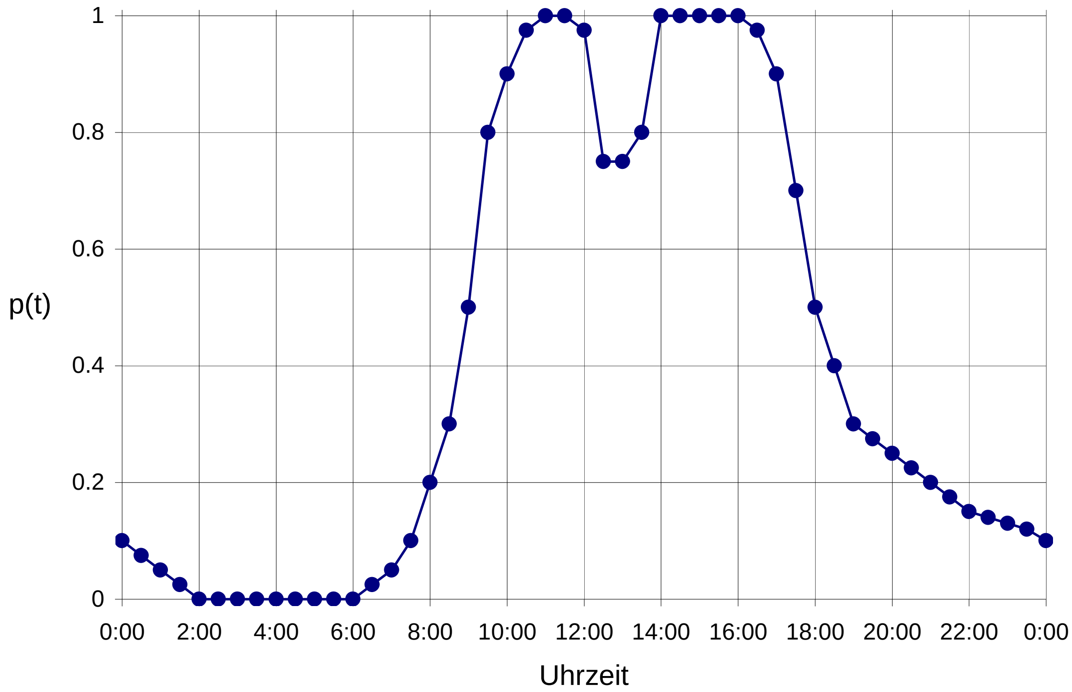

Abbildung 5.6: Akzeptanzfunktion zur Realisierung des inhomogenen Poisson-Prozesses

der Ausgabegeräte und immer mobileren Benutzern nicht mehr auszugehen. Zur Erzeugung der Marken wurde zunächst eine diskrete Verteilung der Medientypen bestimmt. Mit Wahrscheinlichkeit m_i wird einer Anfrage der Medientyp i zugewiesen. Die $q_{(i,j)}$ nehmen somit folgende Werte an:

$$q_{(i,j)} = \frac{m_i}{k}, \quad i = 1, \dots, 7, \quad j = 0, \dots, k-1. \tag{5.11}$$

Hierbei ist k die Anzahl der in der lokalen Umgebung vorhandenen Parzellen.

5.4.1 Erstes Experiment

Zunächst soll die Abhängigkeit der Zuweisung der Medienströme zu den Ausgabegeräten von der Verteilung der Marken untersucht werden. Dazu werden die Parameter des Auswahlalgorithmus wie folgt gesetzt und für die folgenden Versuche unverändert gehalten: Entfernungszeit = 8, Trägheitsschwelle = 30, Ungeduldsschwelle = 180. Bei der Verteilung der Marken wird zu Analysezwecken zunächst von der nichtrealistischen Annahme ausgegangen, daß alle Marken gleichwahrscheinlich sind. Bei $k = 27$ Parzellen und $m_i = \frac{1}{7}$ für alle i ($i = 1, \dots, 7$) ergibt sich nach Gleichung 5.11

$$q_{(i,j)} = \frac{1}{189}, \quad (i,j) \in \mathbb{X}. \tag{5.12}$$

Die Anfragezeitpunkte werden durch einen inhomogenen Poisson-Prozeß bestimmt, der in der oben beschriebenen Weise mit Hilfe der Akzeptanzfunktion $p(t)$ aus einem homogenen Poisson-Prozeß mit $\lambda = \frac{1}{7}$ erzeugt wird. Es wurden mehrfach 21 aufeinanderfolgende Tage simuliert. Obwohl erwartungsgemäß aufgrund der kleineren Gesamtanzahl der simulierten Anfragen größere Differenzen zwischen den verschiedenen Simulationsläufen auftraten als bei den Szenarien des Abschnitts 5.3, können die Daten eines Durchlaufs für die folgenden Interpretationen als repräsentativ betrachtet werden.

Medientyp	Mittlere Wartezeit	Maximale Wartezeit	Varianz	Vorhandene Ausgabegeräte
1	13,702	634,736	2510,470	31
2	13,780	643,074	2394,560	30
3	16,285	640,113	2843,760	25
4	16,808	658,258	3029,820	25
5	0,067	95,793	1,971	68
6	0,219	180,000	7,188	45
7	0,156	59,272	3,952	45

Tabelle 5.7: Experiment 1. Wartezeit nach Medientypen

Tabelle 5.7 zeigt die mittlere und maximale Aufenthaltszeit der Anfragen in der Wartschlange nach Medientypen geordnet, sowie die entsprechenden Varianzen und die Anzahl der für jeden Medientyp zur Verfügung stehenden Ausgabegeräte nach Tabelle 5.5. Bei dieser Simulation trafen in 21 Tagen 107243 Anfragen ein. Die Mittelwerte sind nach Gleichung 4.15 und die Varianzen nach Gleichung 4.16 berechnet. Die Ergebnisse bzgl. der Wartezeit sind hierbei mit der Anzahl der zur Verfügung stehenden Ausgabegeräte und den Bedienraten (vgl. Tabelle 5.6) korreliert. Wäre die diskrete Gleichverteilung der Marken ein adäquates Modell, so ließe sich hieraus eine Unterversorgung mit Ausgabegeräten zur Darstellung der Audiodaten-enthaltenden Medientypen (Typen 1 – 4) erkennen. Die nahezu gleichen Wartezeiten der Typen 1 und 2, bzw. 3 und 4, rühren daher, daß die vorhandenen Ausgabegeräte besonders häufig diese Typen in Kombination unterstützen.

5.4.2 Zweites Experiment

Die im vorherigen Abschnitt gemachte Annahme, daß alle Marken gleichwahrscheinlich sind, ist unrealistisch. Der größte Teil der Anfragen wird sich aus Text- und Bilddokumenten zusammensetzen. Dagegen werden z. B. Radio und Fernsehen nur äußerst selten in einer Büroumgebung angefordert.

Medientyp	1	2	3	4	5	6	7
m_i	0,01	0,05	0,01	0,05	0,75	0,01	0,12

Tabelle 5.8: Verteilung der Medientypen bei neuen Anfragen

Zur Bestimmung der Verteilung der Marken wird daher in diesem und den folgenden Experimenten von den in Tabelle 5.8 angegebenen Werten für die m_i ausgegangen. Damit ist die Verteilung der Marken durch Gleichung 5.11 gegeben, die Parzellenanzahl bleibt unverändert ($k = 27$). Um Vergleiche mit dem ersten Experiment zu ermöglichen, wird $\lambda = \frac{1}{7}$ ebenfalls beibehalten. Es wurden zur Kontrolle mehrfache Simulationsläufe über 21 Tage durchgeführt. Tabelle 5.9 zeigt repräsentativ die Daten eines Durchlaufs, bei dem insgesamt 107163 Anfragen eintrafen.

Medientyp	Mittlere Wartezeit	Maximale Wartezeit	Varianz
1	0,200	125,475	17,928
2	0,335	180,000	29,297
3	0,442	77,294	21,178
4	0,293	169,666	25,362
5	0,090	180,000	4,920
6	0,296	180,000	34,384
7	0,189	180,000	12,819

Tabelle 5.9: Experiment 2. Wartezeit nach Medientypen

Hierbei läßt sich im Vergleich zu Tabelle 5.7 eine deutlich ausgeglichenere Verteilung der mittleren Wartezeiten erkennen, so daß nicht mehr von einem generellen Mangel an Ausgabegeräten für bestimmte Medientypen gesprochen werden kann. Dies zeigt sich auch an den maximalen Wartezeiten. Liegen in Tabelle 5.7 die maximalen Wartezeiten der Medientypen 1 – 4 noch oberhalb der Ungeduldsschwelle, so bildet die Ungeduldsschwelle, welche auf 180 gesetzt ist, in dieser Simulation eine Obergrenze der maximalen Wartezeit. Es sind somit beim Überschreiten der Ungeduldsschwelle immer noch Ausgabegeräte in anderen Parzellen frei, auf denen in diesem Fall die Anfrage dargestellt wird, um den Anfragesteller nicht noch länger warten zu lassen.

5.4.3 Drittes Experiment

In diesem Experiment soll das Verhalten des Systems unter verschiedenen Belastungen untersucht werden. Es wurde dazu eine Versuchsreihe mit wechselnden λ durchgeführt. Alle anderen Parameter wurden konstant gehalten. Die Simulationzeit wurde mit wachsendem λ schrittweise von zwanzig Tagen auf einen Tag reduziert, um die Anzahl der simulierten Anfragen jedes Durchgangs möglichst konstant bei etwa 75.000 zu halten. Das gesamte Experiment benötigte auf einem Intel Pentium III Prozessor mit 550 MHz eine Woche Rechenzeit.

Abbildung 5.7 zeigt die beobachtete mittlere Wartezeit aller Anfragen und Abbildung 5.8 die maximale Anzahl von Anfragen im System während der Simulation, d. h. das Maximum der Summe der Anfragen in der Warteschlange und in den Ausgabegeräten. Dabei zeigt sich, daß die mittlere Wartezeit aller Anfragen exponentiell

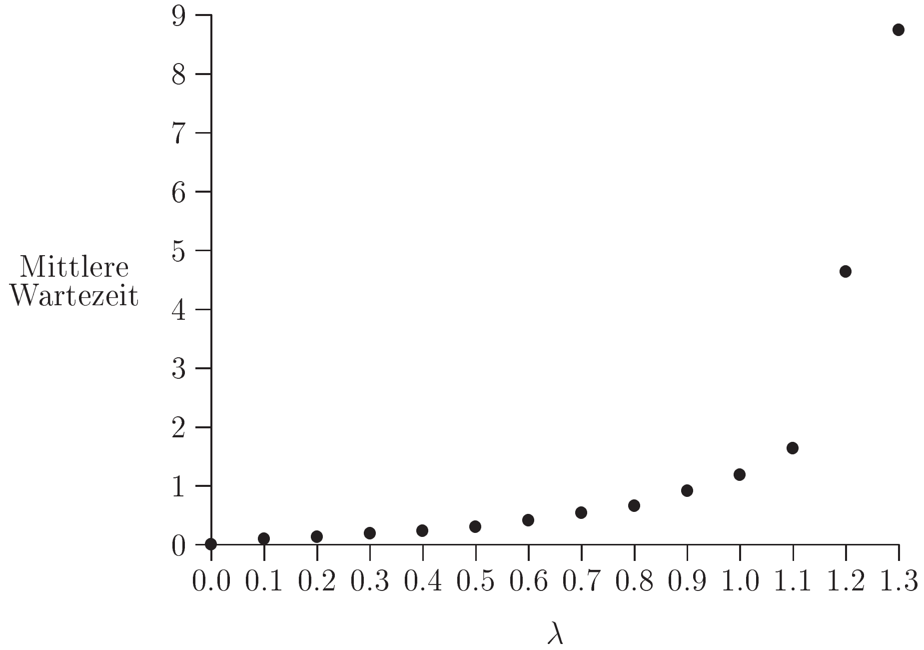

Abbildung 5.7: Experiment 3. Mittlere Wartezeit in Abhängigkeit von λ

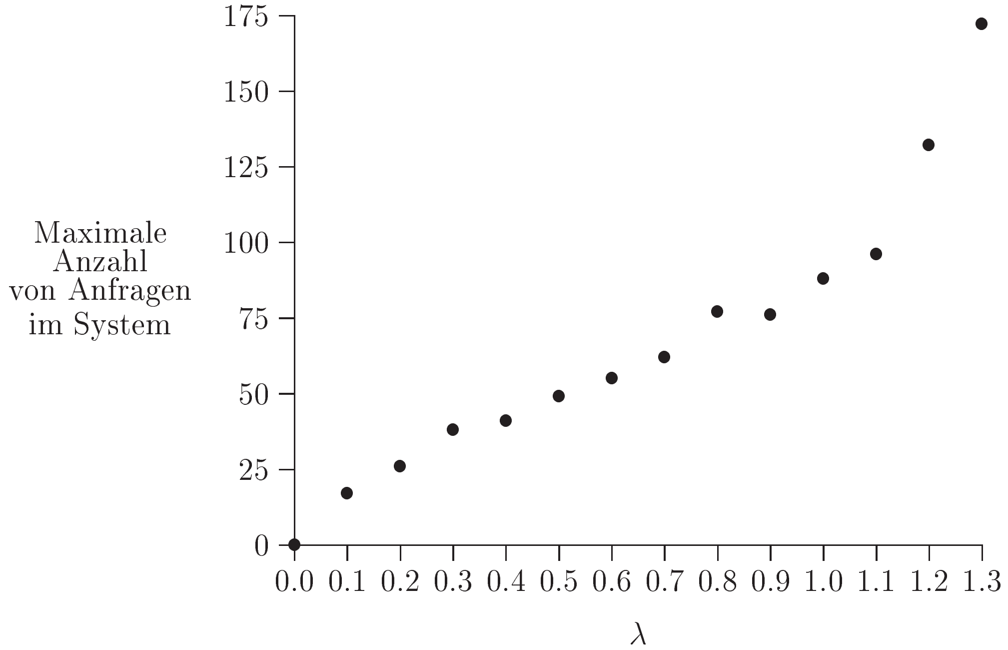

Abbildung 5.8: Experiment 3. Maximale Anzahl von Anfragen im System in Abhängigkeit von λ

wächst. Solange die maximale Anzahl der Anfragen im System unter der Anzahl der vorhandenen Ausgabegeräte (80) bleibt, bleibt auch die mittlere Wartezeit gering. Sobald sich längere Zeit mehr Anfragen im System befinden als Ausgabegeräte vorhanden sind, explodiert erwartungsgemäß die Wartezeit, da das System überlastet ist. Nicht mehr in den Graphiken aufgeführt sind die Werte für $\lambda = 1,4$ (Mittlere Wartezeit = 28,9; Maximale Anzahl = 240) und $\lambda = 1,5$ (Mittlere Wartezeit = 46,1; Maximale Anzahl = 369). Bei diesem exponentiellen Wachstum handelt es sich um ein von einfacheren Bedienungssystemen wie den M/M/r bekanntes Phänomen bei Überlastung. Um die Anzahl der Parzellenwechsel zu minimieren und die Möglichkeiten zur parallelen Ausgabe mehrerer Anfragen innerhalb einer Parzelle zu steigern, ist die Anschaffung weiterer Ausgabegeräte jederzeit sinnvoll, wenn dieses finanziell vom Büro tragbar ist. In dieser Simulation zeigt sich jedoch, daß im TecO mit 80 Ausgabegeräten eine ausreichende Grundversorgung sichergestellt ist. Bei $\lambda = 1$ trifft zu den Spitzenzeiten vor und nach dem Mittagessen im Mittel jede Sekunde eine Anfrage ein, ohne das nennenswerte Wartezeiten entstehen. Eine noch höhere Ankunftsrate ist bei maximal 20 Personen in der Büroumgebung nicht zu erwarten. Daß auch bei zeitweise hohen Ankunftsraten die Medienströme gut verteilt werden, zeigt sich auch daran, daß die maximale Wartezeit bei diesem Experiment für $\lambda = 0$ bis $\lambda = 1,1$ nie die Ungeduldsschwelle (180) überschritt. Erst ab $\lambda = 1,2$ traten Überlastungen auf, die durch diesen Mechanismus nicht mehr abgefangen werden konnten, so daß es vereinzelt zu längeren Wartezeiten kam. Bei einer Implementierung des HermiUC-Systems ist daher die Aufsetzung ein Netzlast-Monitor sinnvoll, der bei einer Überanspruchung eingreift und eventuell Anfragen zurückweist, bzw. auf Schwachstellen im System hinweist. Bei einer ausreichenden Ausstattung mit Ausgabegeräten in der lokalen Umgebung, von der bei der stetig wachsenden Zahl von Computern pro Mensch auszugehen ist, sollte dieses Problem jedoch eher die Ausnahme darstellen.

Die Belastung des Systems im Tagesverlauf zeigt beispielhaft für $\lambda = \frac{1}{2}$ Abbildung 5.9. Hierbei ist die starke Korrelation von $\hat{Q}(t)$ mit der Akzeptanzfunktion des inhomogenen Poisson-Prozesses $p(t)$ (Abbildung 5.6) erkennbar. Die ersten Anfragen treffen morgens nach 6:00 Uhr (21.600 Sek.) ein, die Mittagspause liegt um 12.45 Uhr (45.900 Sek.) herum, dann nimmt die Belastung des System über den späten Nachmittag und Abend hinweg kontinuierlich ab, bis morgens um 2.00 Uhr (7.200 Sek., linker Rand des Diagramms) spätestens die letzten Mitarbeiter das Büro verlassen haben.

5.4.4 Viertes Experiment

In diesem Abschnitt soll der Einfluß der Parameter des Auswahlalgorithmus auf die Verteilung der Medienströme simulationstechnisch untersucht werden. Für jeden Parameter wurden verschiedene Versuchsreihen gefahren, zum einem mit $\lambda = 1$ (Simulationszeit je zwei Tage), was nach Abschnitt 5.4.3 als eine starke Belastung des Systems angesehen werden kann, da zu Spitzenzeiten tagsüber im Mittel jede Sekun-

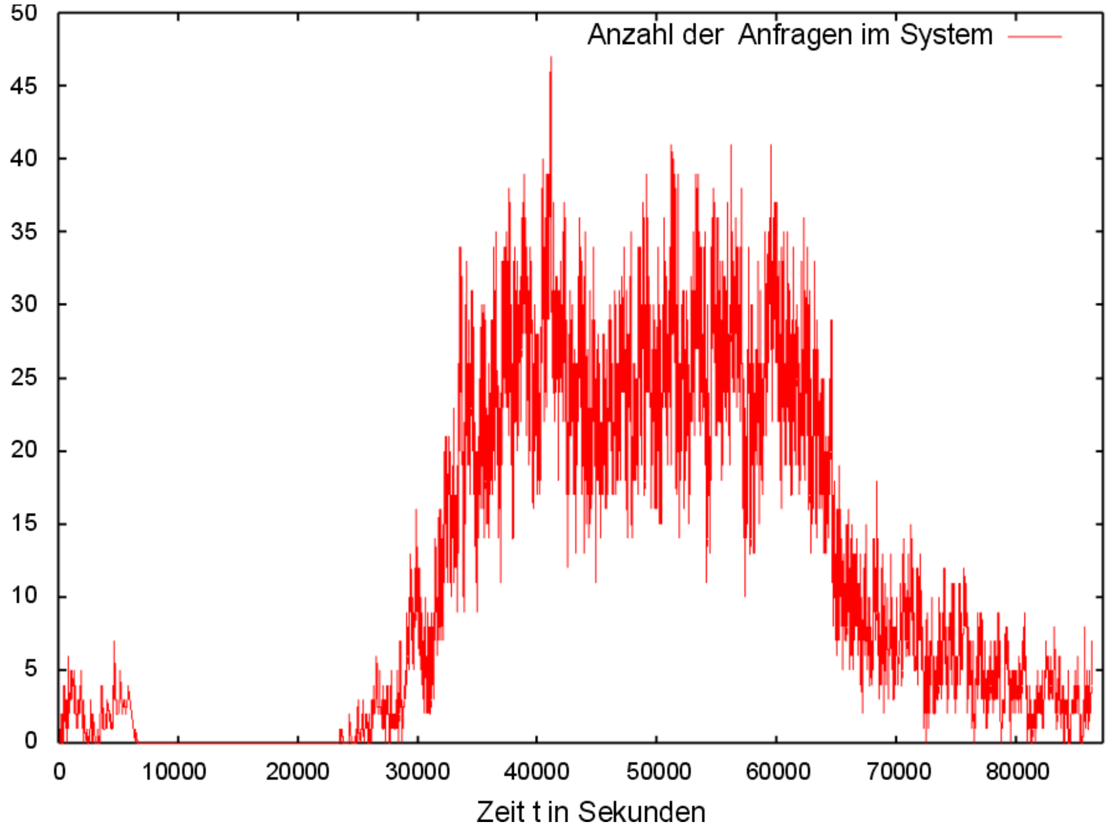

Abbildung 5.9: Experiment 3. Anfragenverteilung über den Tag

de eine Anfrage eintrifft. Zum anderen mit $\lambda = \frac{1}{3}$ (Simulationszeit je sieben Tage), um die Auswirkungen der Parameter bei geringerer Belastung zu betrachten. Es wird auf einige grundlegende Zusammenhänge eingegangen. Eine exakte Interpretation ist dadurch erschwert, daß nicht nur das Zusammenspiel der Parameter eine wichtige Rolle spielt, sondern die Stärke ihrer Einflußnahme auch in nicht vernachlässigbarer Weise von der Topologie und der Verteilung der Medientypen einschließlich ihrer Darstellungslängen abhängt.

5.4.4.1 Variation der Ungeduldsschwelle

Zunächst soll der Einfluß der Ungeduldsschwelle untersucht werden. Dazu wurden für beide gerade angesprochenen λ eine Simulationsreihe umgesetzt, bei der die Ungeduldsschwelle im Intervall von 0 bis 300 in Abständen von 30 variiert wurde. Dabei ergab sich, daß bei geringerer Belastung ($\lambda = \frac{1}{3}$) im Intervall von 0 bis 270 das Maximum der Wartezeiten aller Anfragen immer gleich der Ungeduldschwelle war, bis es anschließend darunter blieb. Die mittlere Wartezeit ist jedoch nur bei Ungeduldsschwellen ≤ 60 verringert, was jedoch mit einer erhöhten relativen Häufigkeit eines Parzellenwechsels erkauft werden muß (vgl. auch Abschnitt 5.3.3). Der Parameter Ungeduldsschwelle erfüllt somit außerhalb des kritischen Bereichs nahe Null zuverlässig seine in Abschnitt 3.2.2.1 angesprochene Funktion, nämlich vereinzelte lange Wartezeiten zu verhindern, ohne sonst die Verteilung wesentlich zu beeinflussen. Bei starker Belastung ($\lambda = 1$) setzt die Gleichheit des Maximums der Wartezeiten mit der Ungeduldsschwelle erst bei größeren Werten der Ungeduldsschwelle ein. Entsprechend bleibt die Gleichheit dann auch länger erhalten, bis die Maximale Wartezeit unter die Ungeduldsschwelle sinkt.

5.4.4.2 Variation der Entfernungszeit

Bei geringer Belastung hat die Entfernungszeit wenig Einfluß auf die Verteilung der Medienströme, da in den meisten Fällen eine Ausgabe ohne Parzellenwechsel erfolgen kann. Bei stärkerer Belastung steigt die mittlere Wartezeit, die relative Häufigkeit eines Parzellenwechsels dagegen sinkt mit wachsender Entfernungszeit. In weiten Bereichen zeigten sich diese Abhängigkeiten als linear. Bei Wahl besonderer Topologien und Änderungen der Verteilung der Medientypen und ihrer Darstellungszeiten treten jedoch auch zeitweise nichtlineare Abhängigkeiten auf. Auf ein solches Beispiel wird im folgenden Abschnitt eingegangen.

5.4.4.3 Variation der Trägheitsschwelle

Mit Vergrößerung der Trägheitsschwelle steigt die mittlere Wartezeit, dafür nimmt die relative Häufigkeit eines Parzellenwechsels zu. Damit läßt sich regulieren, ab welchem Zeitgewinn, was den Darstellungsbeginn betrifft, die Benutzer bereit sind, sich in eine anderen Parzelle zu begeben. Ebenso wie bei der Entfernungszeit gibt es hier nach den Erfahrungen mit vielfachen Simulationsszenarien in den meisten Fällen keine sprunghaften Änderungen. An dieser Stelle soll jedoch zur Vollständigkeit auch ein Ausnahmefall dargestellt werden.

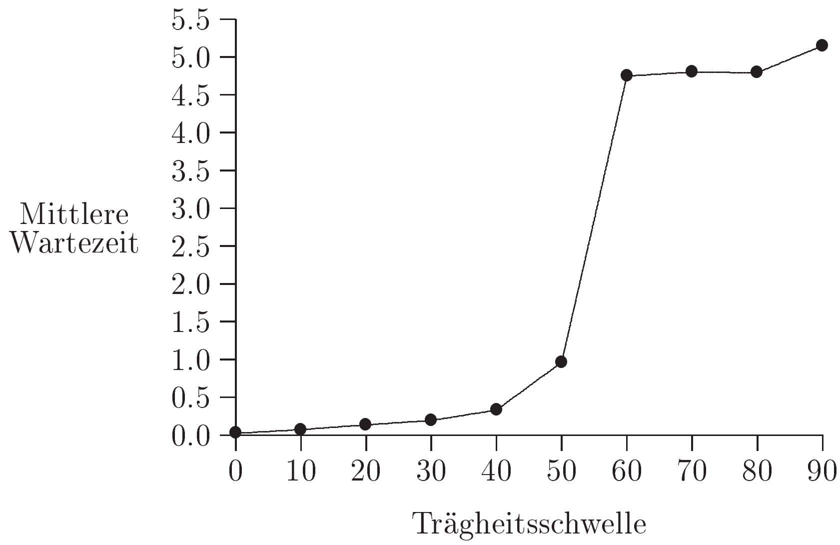

Abbildung 5.10: Experiment 4. Mittlere Wartezeit in Abhängigkeit der Trägheitsschwelle

Abbildung 5.10 zeigt für $\lambda = \frac{1}{3}$ die mittlere Wartezeit in Abhängigkeit der Trägheitsschwelle. Für jeden Punkt wurden sieben aufeinanderfolgenden Tage simuliert. In Abbildung 5.11 ist die korrespondierende relative Häufigkeit eines Parzellenwechsels dargestellt. Dabei ist ein deutlicher Sprung zu erkennen. Durch Veränderung der in Tabelle 5.6 angegebenen Bedienraten läßt sich dieser beseitigen. Derartige Sprünge lassen sich durch die in Algorithmus 3.1 enthaltenden Kriterien mit den Auswahlparametern erklären. Bei gedächtnislosen Verteilungen für die Darstellungszeit kann als geschätzte Wartezeit bis zur Beendigung einer Darstellung zeitlich unbegrenzter Medientypen nur der Erwartungswert der Verteilung verwendet werden. Werden nun die Auswahlparameter variiert, wie in diesem Fall die Ungeduldsschwelle, so ist die Bedingung „geschätzte Wartezeit auf AG in Anfrageparzelle $-$ Entfernungszeit $>$ Trägheitsschwelle“ bei *Trägheitsschwelle* $= 50$ deutlich häufiger erfüllt, als bei *Trägheitsschwelle* $= 60$. Dadurch steigt die mittlere Wartezeit innerhalb diese Intervalls deutlich an und die relative Häufigkeit eines Parzellenwechsels sinkt.

5.5 Fazit

Insgesamt zeigt sich ein stabiles und ausgeglichenes Systemverhalten. Wie bereits in Kapitel 3 erwähnt, wurden schon während des Entwurfs des Auswahlalgorithmus

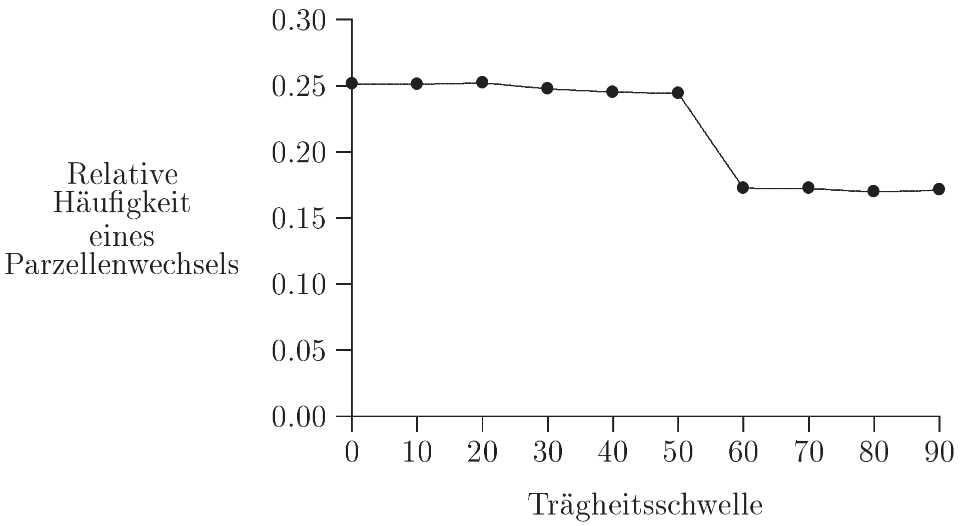

Abbildung 5.11: Experiment 4. Relative Häufigkeit eines Parzellenwechsels in Abhängigkeit der Trägheitsschwelle

erste Simulationen durchgeführt, so daß durch den Arbeitsaufwand für die Simulation erste Schwächen des Entwurfs erkannt und beseitigt werden konnten, bis es zu dem in diesem Kapitel analysierten Systemverhalten kam. Als kritisch ist die große Anzahl von Annahmen zu betrachten, auf denen die Simulation beruht, so daß mit Verallgemeinerungen der Ergebnisse der Büroszenarien auf andere lokale Umgebungen vorsichtig umgegangen werden muß. Dieses zeigte sich besonders im vierten Experiment, bei dem eine umfassenden Analyse aller Zusammenhänge, die zu den teilweise sprunghaften Änderungen der Leistungskenngrößen führen, nicht möglich war. Interessant erscheint an dieser Stelle das Konzept der dynamische Veränderung der Parameter des Auswahlalgorithmus abhängig von der Ankunftsrate und den Darstellungslängen. Dieses könnte weiter zur Stabilität des Systems beitragen. Dafür wurde sich die Nachvollziehbarkeit der Verteilung der Ausgabeströme für Menschen verschlechtern, der Grund, weswegen bisher darauf verzichtet wurde.

6. Implementierung des HermiUC-Systems

Bei der Motivation der Modellierung und Simulation (Abschnitt 4.1) wurde bereits darauf hingewiesen, daß eine Implementierung größeren Maßstabs momentan nur mit erheblichen Zeit- und Kostenaufwand, der den Umfang dieser Arbeit überstiege, realisierbar wäre. Um jedoch zumindest die technische Machbarkeit mit heutigen Mitteln aufzuzeigen, wird in diesem Kapitel eine Implementierungsmöglichkeit und Eingliederung des HermiUC-Systems in existierende Anwendungen des TecOs beschrieben.

Eine Implementierung des HermiUC-Systems besteht aus unterschiedliche Ausgabegeräte (z. B. Fernsehern, Lautsprechern), den persönlichen Eingabegeräten und dem Controller, unterstützt durch ein oder mehrere verschiedene Positionssysteme (Abbildung 6.1). Die Geräte und Systeme sind mittels verschiedener Netzwerktypen miteinander verbunden. Die Eingabegeräte kommunizieren drahtlos mit dem Controller und den Positionssystemen, während die übrigen Komponenten sich mittels drahtgebundener Kommunikation austauschen. Teile der Implementierung können von früheren Projekten am TecO übernommen werden oder zumindest durch Erweiterungen auf diesen aufbauen. Zum Beispiel können das Netzwerk und das Lokationssystem der MediaCup [4] verwendet werden. Ebenso auf Seite der Eingabegeräte Anwendungen der UbicompBrowser- [6] und ElectronicManual-Projekte [3]. Zunächst soll weiter auf die Gerätekomponenten des HermiUC-Systems eingegan-

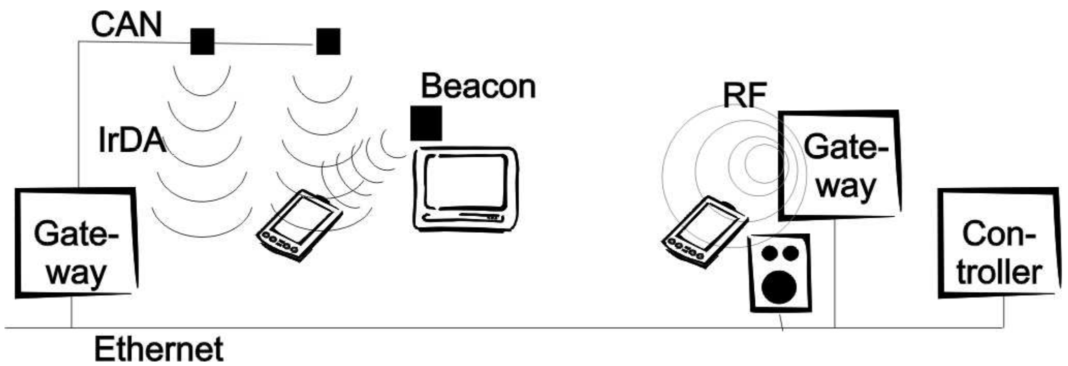

Abbildung 6.1: Schematischer Aufbau einer HermiUC-Implementierung

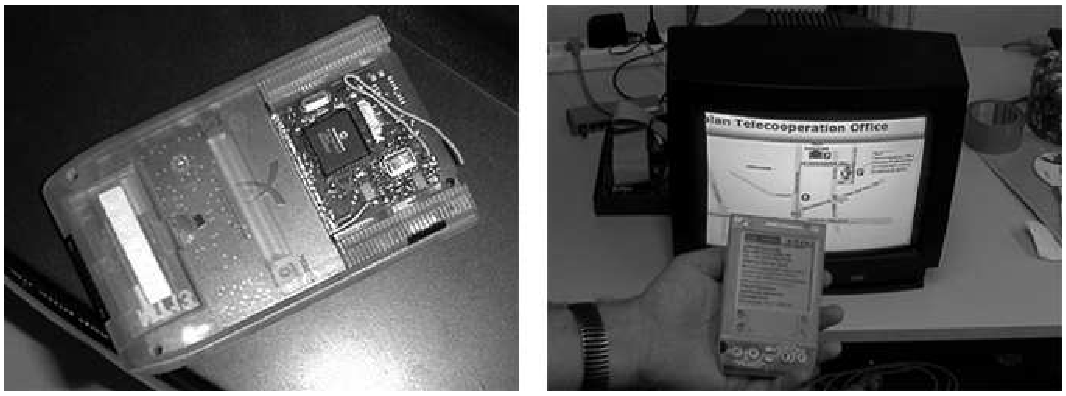

Abbildung 6.2: Prototypische HermiUC-Implementierung. Mobiles Eingabegerät, HermiUC-erweitertes Ausgabegerät

gen werden, bevor anschließend die zur Verfügung stehenden Kommunikations- und Lokationstechnologien besprochen werden.

6.1 Programme und Geräte

Als Eingabegeräte bieten sich PDAs (z. B. PalmPilot oder Handspring Visor) an. Diese werden anhand ihrer jeweiligen IP-Adresse eindeutig identifiziert. Da ihnen diese erst beim Betreten der HermiUC-Umgebung, genauer beim Stellen der ersten Anfrage, automatisch zugeordnet wird und diese nach dem Verlassen, bzw. wenn sich keine Anfragen des Benutzers mehr im System befinden, nicht mehr benötigt wird, lassen sich damit Überwachungsszenarien wie sie bei den Sicherheitsaspekten (Abschnitt 2.1.2) angesprochen wurden, unterbinden. Auf den Geräten können einerseits mittels einer für das HermiUC-System entwickelten Waba-basierten Anwendung explizit Multimedia-Daten ausgewählt und somit das System beauftragt werden. Implizite Anfragen andererseits werden z. B. erzeugt, wenn man mit dem PDA durch das World Wide Web navigiert. Mediendaten, die nicht auf dem PDA dargestellt werden können, werden HermiUC übergeben und somit automatisch auf Ausgabegeräte in der Umgebung umgeleitet. Zur Positionserkennung mittels des PDAs bieten sich verschiedene Verfahren auf Funk- oder Infrarotbasis an.

Als Controller kann ein herkömmlicher PC eingesetzt werden, der eine SQL-Datenbank für die Verwaltung der Ausgabegeräte verwendet. Wie bei der Analyse in Abschnitt 2.2.1 bereits angesprochen wurde, kann der Controller mittels des Ereignisbasierten RAUM-Protokolls [10] den Ausgabegeräten Aufträge zur Darstellung erteilen. Um Ausgabegeräte wie Fernseher und Radios, die nicht standardmäßig über einen Netzwerkanschluß und die notwendigen Kommunikationsfähigkeiten verfügen, in das HermiUC-System zu integrieren, werden diese mit kleinen PC104-Platinen erweitert, auf denen das Betriebssystem Windows 95 läuft (Abbildung 6.2, rechts), während auf den anderen PCs, die selber als Ausgabegerät dienen oder elektronische

Wandtafeln kontrollieren, Windows NT installiert ist. Auf jedem PC muß ein Dienst aktiv sein, der ihm erlaubt, über das Netzwerk mittels des RAUM-Protokolls zu kommunizieren, die Mediendaten zu laden und sie auf dem Bildschirm oder über die angeschlossenen Lautsprecher wiederzugeben. Die Darstellung läßt sich mit Hilfe der Microsoft Internet Explorer Control realisieren. Damit werden alle Medienformate unterstützt, die dem Internet Explorer direkt oder mittels Plug-Ins bekannt sind. Ein Prototyp dieses Darstellungsdienstes wurde im Zuge dieser Arbeit in Microsoft Visual Basic 6.0 implementiert.

6.2 Netzwerk und Lokalisation

Die Ausgabegeräte können mit dem HermiUC-Controller durch UDP/IP über Ethernet verbunden werden und mittels des RAUM-Protokoll kommunizieren. Das Netzwerk wird ebenso zum Laden der Medienströme verwendet. Hierbei steht eine Vielzahl von Protokollen zur Verfügung (z. B. RealPlayer oder HTTP). Die Verbindung zwischen den Eingabegeräten und dem Controller kann auf verschiedene Weisen erfolgen. Eine Möglichkeit besteht im Einsatz von Infrarotkommunikation (IrDA) auf Basis der MediaCup-Umgebung. Dabei kann die in Palm OS basierten Geräten eingebaute IrDA-Schnittstelle verwendet werden. Anfragen von den Eingabegeräten werden durch eine Infrastruktur von IrDA-Empfängern verbunden mit einem Car Area Network (CAN) bis zum Ethernet getunnelt, mit dem der Controller und die Ausgabegeräte verbunden sind. Eine alternative Möglichkeit besteht in der Verwendung von Funk (Radio Frequency, RF). Auch in diesem Fall findet mittels eines Umsetzers (Gateways) die Weiterleitung auf ein IP-basiertes Ethernet statt (vgl. Abbildung 6.1). Für diese Variante wurde am TecO ein Springboard-Modul für den Handspring Visor gebaut (Abbildung 6.2, links), das Entfernungen bis zu 30 Meter unterstützt. Das Modul wurde auf Basis eines Radio Frequency Monolithics (RFM) Sendeempfängers entwickelt. Es bietet eine Baudrate von 128 kbit/s bei 868 MHz auf einem Kanal. Das dazu entwickelte Kommunikationsprotokoll erlaubt eine Skalierung mittels Software von 50 Zentimetern bis zu 30 Metern und verbraucht im Mittel 2 mA [5]. Weiterhin ist es ebenfalls in der Lage Kommunikationspartner mit einer mittleren Granularität von einem Meter zu lokalisieren.

Die Lokalisierung der Eingabegeräte kann z. B. durch Messung der Stärke der RF-Signale oder durch in den Parzellen installierte IrDA-Sender (Beacons) erfolgen, wie in Abbildung 6.1 schematisch dargestellt ist. Diese Sender können eine Standortinformation periodisch oder auf Anforderung aussenden. Der Vorteil dabei liegt darin, daß die Verarbeitung der Lokationsinformation auf den Eingabegeräten stattfindet, welche diese Information nur an den Controller weitergeben, wenn sie wirklich benötigt werden. Damit lassen sich die Implikationen auf den Entwurf bzgl. der Sicherheitsaspekte (Abschnitt 2.1.2) umsetzen. Die Positionsdaten gehören dem Benutzer, er entscheidet, wann sie dem System mitgeteilt werden, eine kontinuierliche Überwachung der Aufenthaltsorte der Benutzer im System ist damit nicht möglich.

7. Zusammenfassung und Ausblick

In diesem Kapitel wird zunächst eine Zusammenfassung der Ergebnisse und Erkenntnisse dieser Arbeit gegeben. Anschließend erfolgt ein Ausblick auf mögliche nachfolgende Arbeiten. Dieser gliedert sich in zwei Abschnitte. Im Ersten werden mögliche weitere mathematische Untersuchungen dargestellt. Im Zweiten werden Erweiterungen des technischen Entwurfs besprochen, auf die in dieser Arbeit teilweise mit Rücksicht auf ein handhabbares mathematisches Modell verzichtet wurde.

7.1 Zusammenfassung

Ziel dieser Arbeit war es, ein System zur automatischen Verteilung von Ausgabeströmen in Ubiquitous Computing Umgebungen zu entwickeln. Nur dadurch kann bei wachsenden Rechneranzahlen pro Mensch gewährleistet werden, daß die vorhandenen Ausgabekapazitäten der die Benutzer umgebenden Geräte voll zur Verfügung stehen und ohne wachsenden Verwaltungsaufwand auf Seiten des Menschen genutzt werden können.

Dazu wurde zunächst der Umgang des Menschen mit heutigen Computern untersucht. Daraus ergab sich, daß eine möglichst einfache Bedienbarkeit durch den Menschen ein zentrales Kriterium für die Akzeptanz und Güte eines solchen Systems ist. Weiterhin wurden Medien und der Umgang des Menschen mit diesen analysiert. Dies führte sowohl zu einer Klassifizierung von Medientypen, auf der der nachfolgende Entwurf aufbaute, als auch zu Implikationen auf den Entwurf bzgl. der Verteilung und Darstellung der Multimedia-Daten.

Anschließend wurde der Gesamtaufbau des entwickelten Systems beschrieben. Es setzt sich aus mobilen persönlichen Eingabegeräten, dem Controller als Verwaltungseinheit und stationären Ausgabegeräten zusammen. Eine wichtige Funktionskomponente ist die Verarbeitung der Lokationsinformationen. Hier wurde ein zweistufiges Verfahren entwickelt und zur algorithmischen Verarbeitung graphentheoretisch abstrahiert. Die Zuordnung von Benutzeranfragen zu Ausgabegeräten wird mittels eines Algorithmus vorgenommen, der als Grundkriterien die technische Eignung, den Belegungszustand, die Typspezialisierung und die Darstellungsqualität der vorhandenen Ausgabegeräte verwendet. Zuletzt wurde der Kommunikationsablauf des Gesamtsystems beschrieben.

Das entwickelte System wurde durch ein mathematisches Modell beschrieben. Dabei handelt es sich um ein offenes Warteschlangennetz. Der Ankunftsstrom der Benutzeranfragen ist durch einen inhomogenen markierten Poisson-Prozeß modelliert. Die Marken bestimmen dabei die Medientypen, die Anfrageorte und die Darstellungslängen, welche abhängig vom Typ mit verschiedenen Parametern exponentialverteilt sind.

Anhand des mathematischen Modells wurde eine Diskrete-Ereignis-Simulation unter OMNeT++ programmiert. Es wurden zunächst einfache Szenarien des Systems simuliert und analysiert um die Korrektheit der Simulation zu validieren. Anschließend wurde die Büroumgebung des TecOs mit wechselnden Parametern simuliert und es wurden Aussagen über die Güte der Verteilung der Medienströme gemacht. Hierbei zeigte sich ein stabiles Systemverhalten, jedoch erwiesen sich in einige Spezialfällen die Leistungskenngrößen als nicht glatt abhängig von den Systemparametern.

Zum Abschluß wurde eine mögliche Implementierung und Eingliederung des Systems in eine Ubiquitous Computing Umgebung dargestellt, die als weiterführende Arbeit realisiert werden kann.

7.2 Ausblick

7.2.1 Weiterführende mathematische Untersuchungen

Das Gesamtsystem verschließt sich offenbar einer induktiven mathematischen Untersuchung weitestgehend. Jedoch finden sich auch unter den Spezialfällen des Systems mit nur zwei Ausgabegeräten und höchstens zwei Parzellen, wie sie z. B. in den Abschnitten 5.3.2 und 5.3.3 behandelt wurden, bereits viele bisher in der Literatur nicht untersuchte interessante Bedienungssysteme, aus deren genauer Analyse man weitere Rückschlüsse auf das Verhalten des Gesamtsystems ziehen könnte. Auch die Untersuchung weiterer komplexerer Topologien könnte neue Aufschlüsse zur Übertragbarkeit der hier dargestellten Resultate bringen.

7.2.2 Erweiterungen des technischen Entwurfs

Bei einer Implementierung könnte das System um einige Dinge erweitert werden, auf die in dieser Arbeit mit Rücksicht auf ein handhabbares mathematisches Modell verzichtet wurde. Dabei wären zunächst mobile Ausgabegeräte zu nennen, was weiterhin zu dynamischen Parzellen führen würde. Die Kontextverarbeitung sollte erweitert werden, um z. B., wie im Anwendungsszenarium und Abschnitt 2.2.1 beschrieben, automatisch Besprechungssituation zu erkennen und in diesem Fall großflächige Ausgabegeräte zu wählen. Ebenso sollte die Möglichkeit untersucht werden, wie Darstellungen bzw. Anwendungen mit Hilfe eines solchen System dem Benutzer in andere Räume folgen können, bzw. wie eine nicht unterbrechungsfreie Bedienung umgesetzt werden kann, die es Benutzern erlaubt, zu früheren Anfragen zurückzukehren.

7.3 Danksagung

Danken möchte ich meinen Betreuern Herrn Prof. Dr. G. Last und Herrn Dr. M. Beigl, ohne deren Hilfe die Diplomarbeit in dieser Form nicht möglich gewesen wäre. Weiterhin gilt mein Dank Herrn A. Schmidt für die Diskussion programmiertechnischer Fragen und Herrn Priv. Doz. Dr. D. Kadelka für die Erörterungen zur Warteschlangentheorie.

A. OMNeT++ NED Programmkode

A.1 Deklarationen der Simple Modules

```
//-------------------------------------------------------------
// File: szene1mod.ned
//
// Declaration of simple module types for HermiUC-Simulation
//
// Author: Andreas Neumann (TecO, University of Karlsruhe)
//-------------------------------------------------------------


// Generator --
// generates mediarequests
simple Generator
    parameters:
        num_messages : numeric,
        gen_speed : numeric,
        media_type : numeric,
        parzelle : numeric,
        inhomogen : bool,
        media_uniform : bool,
        view_length_1, view_length_2, view_length_3, view_length_4,
        view_length_5, view_length_6, view_length_7 : numeric;
    gates:
        out: to_contr;
endsimple


// universelles Ausgabegeraet
simple Ausgabe
    parameters:
     belegt : bool,
        parzelle : numeric,
        medientypen : numeric;
```

```
    gates:
        in: from_contr;
        out: to_contr;
endsimple


// System Controller
simple Controller
    parameters:
        num_parzellen : numeric const,
        not_moving_time: numeric const,
        impatience_time: numeric const,
        parzellen_walking_time: numeric const,
        expectation_1, expectation_2, expectation_3, expectation_4,
        expectation_5, expectation_6, expectation_7 : numeric;
    gates:
        in: from_generator;
        in: from_ausgabe[];
        out: to_ausgabe[];
endsimple


// Controller-Ausgabegeraet Verbindung
channel Signalisierung
    delay 0.0
    error 0.0
endchannel
```

A.2 Netzwerkbeschreibung

```
//-----------------------------------------------------------
// File: szene1.ned
//
// Network description of HermiUC-Simulation
//
// Authors: Andreas Neumann (TecO, University of Karlsruhe)
//------------------------------------------------------------

import "szene1mod";

//
// Szene
//
```

```
module Szene
    parameters:
        num_ausgabe : numeric const;
    submodules:
        ausgabe: Ausgabe[num_ausgabe];
            parameters:
                belegt = false,
                parzelle = input,
                medientypen = input;
            display: "p=194,75;i=screen_passive";
        controller: Controller;
            parameters:
                num_parzellen = input,
                not_moving_time = input,
                impatience_time = input,
                parzellen_walking_time = input,
                expectation_1 = input,
                expectation_2 = input,
                expectation_3 = input,
                expectation_4 = input,
                expectation_5 = input,
                expectation_6 = input,
                expectation_7 = input;
            gatesizes:
                from_ausgabe[num_ausgabe],
                to_ausgabe[num_ausgabe];
            display: "p=196,199;i=pc";
        generator: Generator;
            parameters:
                num_messages = input,
                gen_speed = input,
                media_type = input,
                parzelle = input,
                inhomogen = input,
                media_uniform = input,
                view_length_1 = input,
                view_length_2 = input,
                view_length_3 = input,
                view_length_4 = input,
                view_length_5 = input,
                view_length_6 = input,
                view_length_7 = input;
            display: "p=197,310;i=gen";
    connections:
```

```
        generator.to_contr --> Signalisierung
                           --> controller.from_generator;
        for i=0..num_ausgabe-1 do
            ausgabe[i].to_contr --> Signalisierung
                                --> controller.from_ausgabe[i];
            ausgabe[i].from_contr <-- Signalisierung
                                  <-- controller.to_ausgabe[i];
        endfor;
    display: "p=66,10;b=437,367,rect";
endmodule


//
// szene1
//
// instance of the HermiUC-Simulation
//
network szene1 : Szene
    parameters:
        num_ausgabe = input(80, "Anzahl Ausgabegeraete");
endnetwork
```

B. OMNeT++ C++ Programmkode

B.1 generator.hh

```
//————————————————————————————————————————
// File: generator.hh
//
// Declaration of simple module types: Generator
//
// Author: Andreas Neumann (TecO, University of Karlsruhe)
//————————————————————————————————————————


#include <stdio.h>
#include <set>
#include "omnetpp.h"


class Generator : public cSimpleModule
{
  Module_Class_Members(Generator,cSimpleModule,16384);
  virtual void activity();
  double intensitaetsf(simtime_t t);
};
```

B.2 generator.cc

```
//————————————————————————————————————————
// File: generator.cc
//
// Implementation of simple module types: Generator
//
// Author: Andreas Neumann (TecO, University of Karlsruhe)
//————————————————————————————————————————
```

```
#include "generator.hh"
#include "controller.hh"
#include <Binomial.h>
#include <ACG.h>


// Turn on code that prints debug messages
#define TRACE_MSG


// Module registration:
Define_Module( Generator );


// Activity function of the simple module
void Generator::activity()
{
  ev << "Generator startet. Module ID: " << id()
     << endl;

  int num_messages = par("num_messages");
  cPar& gen_speed = par("gen_speed");
  cPar& media_type = par("media_type");
  cPar& parzelle = par("parzelle");
  char msgname[32];
  int chance;
  double uniform_result;
  double media_distr_limit;

  double media_type_distr[7]
    = { 0.01, 0.05, 0.01, 0.05, 0.75, 0.01, 0.12+0.1};
  //+0.1 Sicherheitszuschlag

  ACG gen(73,55);
  Binomial binomi(1, 0.5, &gen);

  for (int i = 0; i < num_messages; i++)
    {
      wait( (simtime_t) gen_speed );

      if (par("inhomogen"))
        {
          binomi.u(intensitaetsf(simTime()));
#ifdef TRACE_MSG
```

```
            ev << "Inhomogener Poissonprozess" << endl
               << "Binomial p: " << binomi.u() << endl;
#endif
            if (!binomi())
              {
#ifdef TRACE_MSG
                ev << "Event wird verworfen." << endl;
#endif
                i--;
                continue;
              }
          }

        sprintf( msgname, "request-%d", i);

        cMessage *msg = new cMessage( msgname );
        msg->addPar("request_parzelle") = (int) parzelle;
        if (par("media_uniform"))
          {
            msg->addPar("media_type") = (int) media_type;
          }
        else
          {
#ifdef TRACE_MSG
            ev << "Not media_uniform" << endl;
#endif
            uniform_result = genk_uniform( 10, 0.0, 1.0);
#ifdef TRACE_MSG
                ev << "uniform_result = " << uniform_result
                   << endl;
#endif
            media_distr_limit = 0.0;
            for (int media_iter = 1; media_iter <= 7;
                  media_iter++)
              {
                media_distr_limit
                  += media_type_distr[media_iter-1];
#ifdef TRACE_MSG
                ev << "media_iter = " << media_iter
                   << endl;
                ev << "media_distr_limit = "
                   << media_distr_limit
                   << endl;
#endif
```

```
                if (uniform_result <= media_distr_limit)
                  {
                    msg->addPar("media_type") = media_iter;
                    break;
                  }
              }
          }
        msg->setTimestamp();

        switch ((int) msg->par("media_type"))
          {
          case 1:
            msg->addPar("view_length")
              = (simtime_t) par("view_length_1");
            break;
          case 2:
            msg->addPar("view_length")
              = (simtime_t) par("view_length_2");
            break;
          case 3:
            msg->addPar("view_length")
              = (simtime_t) par("view_length_3");
            break;
          case 4:
            msg->addPar("view_length")
              = (simtime_t) par("view_length_4");
            break;
          case 5:
            msg->addPar("view_length")
              = (simtime_t) par("view_length_5");
            break;
          case 6:
            msg->addPar("view_length")
              = (simtime_t) par("view_length_6");
            break;
          case 7:
            msg->addPar("view_length")
              = (simtime_t) par("view_length_7");
            break;
          }

#ifdef TRACE_MSG
        ev << "Generating " << msgname
           << ", media_type "
```

```
          << (long) msg->par("media_type")
          << ", request_parzelle "
          << (long) msg->par("request_parzelle")
          << ", view_length "
          << (simtime_t) msg->par("view_length")
          << " sec." << endl;
#endif

      send( msg, "to_contr" );
    }
}

// Acceptancefunction for the inhomogeneous Poisson-process
double Generator::intensitaetsf(simtime_t t)
{
  double p=0.0;
  long sim_time, interval_time, interval;
  double p_point[49]=
  {
    .1, .075, .05, .025, 0, 0, 0, 0, 0, 0, 0, 0,
    0, .025, .05, .1, .2, .3, .5, .8, .9, .975, 1, 1,
    .975, .75, .75, .8, 1, 1, 1, 1, 1, .975, .9, .7,
    .5, .4, .3, .275, .25, .225, .2, .175, .15, .14, .13,
    .12, .1
  };

  sim_time = long(t);
  sim_time = sim_time % 86400; // Modulo 24 h
  interval_time = sim_time % 1800; // Modulo 30 min
  interval = sim_time / 1800; // Div 30 min

  p = p_point[interval]
    + (p_point[interval+1]-p_point[interval])
    *(double(interval_time)/1800.0);

  return p;
}
```

Literaturverzeichnis

[1] Abowd, G. D.: *Ubiquitous Computing: Research Themes and Open Issues from an Applications Perspective*, GVU Center, Georgia Institute of Technology, Technical Report, GIT-GVU-96-24, 1996

[2] Adler, R. J., Feldman, R. E., Taqqu, M. S.: *A Practical Guide to Heavy Tails*, Birkhäuser, 1998

[3] Beigl, M.: *ElectronicManual*, Human-Computer Interaction: Communications, Cooperation and Application Design, Vol2, H.-J. Bullinger and J. Ziegel ed., Lawrence Erlbaum Associates, 1999

[4] Beigl, M., Gellersen, H.-W., Schmidt, A.: *MediaCups: Experience with Design and Use of Computer-Augmented Everyday Objects*, Computer Networks, Special Issue on Pervasive Computing, Elsevier, Vol. 35, No. 4, März 2001

[5] Beigl, M., Ringwald, M.: *Communication Issues in Ubiquitous Computing*, Proceedings of the 1st International Conference on Internet Computing, Las Vegas, USA, Juni 2001

[6] Beigl, M., Schmidt, A., Lauff, M., Gellersen, H.-W.: *The UbicompBrowser*, Proceedings of the 4th ERCIM Workshop on User Interfaces for All, Stockholm, Schweden, Oktober 1998

[7] Faller, S.: *Analyse eines M/M/2-Bedienungssystems mit Prioritäten*, Diplomarbeit, Fakultät für Mathematik, Universität Karlsruhe (TH), 2000

[8] Fox, A., Johanson, B., Hanrahan, P., Winograd, T.: *Integrating Information Appliances into an Interactive Workspace*, IEEE Computer Graphics and Applications, May/June 2000

[9] Harter, A., Hopper, A.: *A Distributed Location System for the Active Office*, IEEE Network, Vol.8, No. 1, January 1994

[10] Hupfeld, F., Beigl, M.: *Spatially aware local communication in The RAUM system*, Proceedings of the IDMS, Enschede, Niederlande, Oktober 17-20, 2000

[11] Kidd, C. D., Orr, R., Abowd, G. D., Atkeson, C. G., Essa, I. A., MacIntyre, B., Mynatt, E., Starner, T. E., Newstetter, W.: *The Aware Home: A Living*

Laboratory for Ubiquitous Computing Research, Proceedings of the Second International Workshop on Cooperative Buildings, Oktober 1999

[12] Last, G., Brandt, A.: *Marked Point Processes on the Real Line*, Springer, 1995

[13] Norman, D. A.: *The Invisible Computer*, The MIT Press, 1998

[14] Ottmann, T., Widmayer, P.: *Algorithmen und Datenstrukturen*, BI Wissenschaftsverlag, 2. Auflage, 1993

[15] Rolski, T., Schmidli, H., Schmidt, V., Teugels, J.: *Stochastic Processes for Insurance and Finance*, John Wiley & Sons, 1999

[16] Schmidt, A., Beigl, M., Gellersen, H.-W.: *There is more to Context than Location*, Proceedings of the International Workshop on Interactive Applications of Mobile Computing (IMC), Rostock, Germany, November 1998

[17] Stoer, J., Bulirsch, R.: *Numerische Mathematik 1*, Springer, 3. Auflage, 1990

[18] Varga, A.: *OMNeT++, Discrete Event Simulation System*, http://www.hit.bme.hu/phd/vargaa/omnetpp.htm, 2001

[19] Varga, A.: *OMNeT++, Version 2.0, User Manual*, http://www.hit.bme.hu/phd-/vargaa/omnetpp.htm, 2001

[20] Walrand, J.: *An Introduction to Queueing Networks*, Prentice-Hall, 1988

[21] Weiser, M.: *Ubiquitous Computing*, http://www.ubiq.com/hypertext/weiser-/UbiHome.html, 2001

[22] Weiser, M.: *The Computer of the 21st Century*, Scientific American (International Edition), 265(3):94-104, September 1991

[23] Weiser, M.: *Some Computer Science Issues in Ubiquitous Computing*, Communication of the ACM, Volume 36, Number 7, July 1993