Universität Leipzig

Fakultät für Mathematik und Informatik

Institut für Informatik

Web-basiertes Management von Daten aus Online-Anträgen von Bürgern an Verwaltungen am Beispiel einer Antragsbearbeitung für eine Gemeinde

Diplomarbeit

Aufgabenstellung und Betreuung:

Prof. Dr. K. Irmscher

Dipl.-Math. J. Hotzky

F. Gebhardt / S. Kahlert (debis Systemhaus)

vorgelegt von:

Daniel Heinze

Leipzig, Februar 2002

Vorwort

Im Rahmen dieser Arbeit soll ein System entwickelt werden, das möglichst ohne größere Investitionen in neue Hard- und Software das web-basierte Management von Daten aus Online-Anträgen von Bürgern an Verwaltungen ermöglicht. Als Anwendungsbeispiel wird die Antragsbearbeitung in einer Gemeinde betrachtet.

Folgenden Personen möchte ich danken:

- Herrn Prof. Dr. K. Irmscher für die Aufgabenstellung und die Unterstützung der Arbeit,
- Herrn J. Hotzky, Herrn F. Gebhardt und Herrn S. Kahlert für die Betreuung der Arbeit.

Inhaltsverzeichnis

1 Einleitung

1. 1 Motivation

Das Informationszeitalter geht auch an den Kommunen in Deutschland nicht vorbei. Viele von ihnen präsentieren sich im *WorldWideWeb* (*WWW*). Sie verfolgen damit unterschiedliche Ziele. So wollen die Einen Investoren anlocken, indem sie Wirtschaftsstandorte anpreisen, andere hingegen wollen ihre touristische Attraktivität darstellen und zeigen auf ihren Seiten wunderschöne Landschaftsbilder und Beschreibungen der Sehenswürdigkeiten und wieder andere versuchen, ihren Bürgern nützliche Informationen anzubieten.

Dass solche Informationen immer mehr im WWW gesucht werden, beweisen nicht nur die steigenden Internet-Nutzerzahlen, sondern auch die explosionsartig angestiegene Zahl von lokalen Informationsanbietern im WWW und deren Zugriffszahlen (vgl. [KON01]). Laut einer von *Taylor Nelson Sofres* (*TNS*) in 27 Staaten durchgeführten Studie zum Vergleich der E-Government-Aktivitäten (also die Durchführung von Verwaltungsvorgängen auf elektronischem Weg) sind Online-Behördengänge vor allem in Skandinavien weit verbreitet (vgl. Abb. 1-1), während Deutschland nur im Mittelfeld liegt. Der Studie nach liegt der Schwerpunkt in Deutschland vor allem auf der Bereitstellung von Informationen und dem Ausdruck von Formularen. Als hinderlich für bundeseinheitliche E-Government-Systeme erweisen sich die im Grundgesetz verankerten Prinzipien des Föderalismus und der kommunalen Selbstverwaltung (vgl. [CZ4801], S. 12). Dadurch kann nicht garantiert werden, dass Bürger mit Verwaltungen in anderen Bundesländern problemlos online kommunizieren können (zum Beispiel bei Umzügen). Trotzdem hat die Bundesregierung das Ziel, bis 2005 alle Verwaltungsvorgänge des Bundes online abzubilden, während der Deutsche Städte- und Gemeindebund nur gut zwei Drittel der Geschäftsprozesse innerhalb der nächsten drei Jahre online anbieten will.

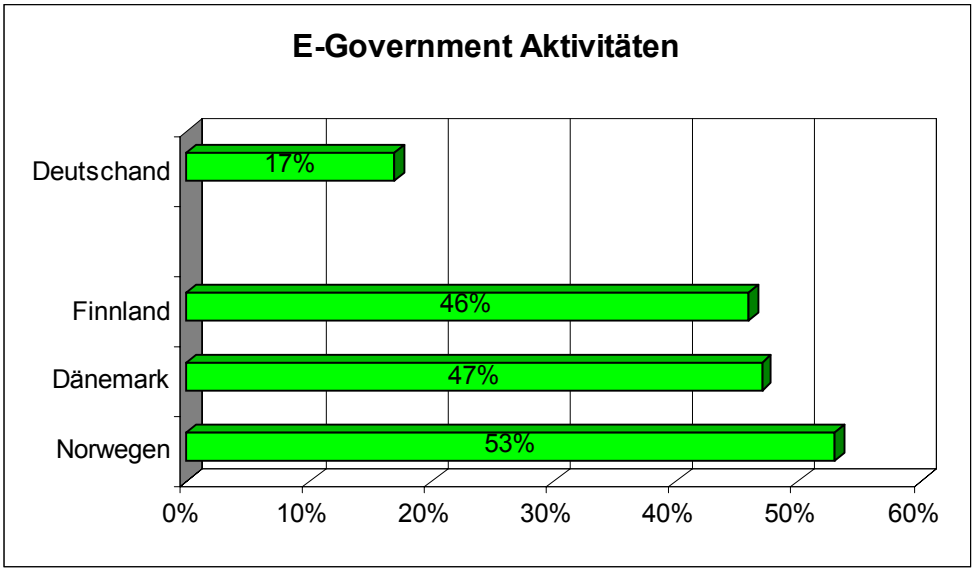

Abb.1-1: E-Government Aktivitäten nach TNS Studie

Ein Problem, welches des Öfteren beim Betrachten der Internet-Seiten auftaucht, sind die veralteten Inhalte. Häufig findet man noch Termine und Ankündigungen vom letzten Jahr in den Bereichen "Aktuelles". Zu erklären ist dies durch den hohen Aufwand, der nötig ist, um eine Internet-Präsenz auf dem aktuellen Stand zu halten. Es gibt zwar Software-Produkte verschiedener Hersteller, mit denen man einfach einen Internet-Auftritt "zusammenklicken" kann, ohne dass man Kenntnis von der darunterliegenden Sprache *Hypertext Markup Language* (*HTML*) hat, aber es ist immer noch aufwändig, die Seiten zu aktualisieren. So muss man beispielsweise erst das entsprechende Dokument finden, das man ändern möchte.

Um dies zu vereinfachen, setzt man mehr und mehr auf dynamisch erzeugte HTML-Seiten. Bei diesen wird der Inhalt, der letztendlich vom Browser ausgegeben werden soll, zur Ladezeit aus einer Datenbank gelesen und in HTML formatiert ausgegeben. Diese Variante ermöglicht es, benutzerfreundliche Masken zu erstellen, mit deren Hilfe es auch HTML-Unkundigen möglich ist, Daten auf dem aktuellen Stand zu halten. Diese über die einfachen Masken eingegebenen Daten werden in der Datenbank gespeichert, von der auch der Webserver seine Daten bezieht. Diese Vorgehensweise wird schon oft praktiziert. So zum Beispiel in dem von Steinkopf und Wiebigke entwickelten Internet-Auftritt der Stadt Meißen (vgl. [Ste99], [Wie99] und [MEI01]). Weiterhin kommen viele Kommunen dem Wunsch der Bürger nach mehr Interaktivität über den Internet-Auftritt der Stadt nach. So können schon heute Formulare und Anträge wie auf der Ho-

mepage der Stadt Halle [HAL01] über das Internet auf den Rechner des Bürgers zu Hause kopiert, dort ausgefüllt und dann per Post an die Stadtverwaltung geschickt werden. Solche Anträge sind zum Beispiel im Falle des Online-Auftritts der Stadt Halle [HAL01]: „Widerspruch gegen die Weitergabe von Daten“, „Antrag auf Erteilung einer einfachen Melderegisterauskunft“ und „Antrag auf Ausstellung einer Ersatz-Lohnsteuerkarte“. Nachteil bei diesen Formularangeboten ist vor allem das Format. Die Anträge werden im Microsoft Word Format zum Download angeboten. Das Ausfüllen der Formulare setzt also das Vorhandensein von Microsoft Word auf dem Rechner des Bürgers voraus. Somit können Nutzer anderer Betriebssysteme, für die es keine Version dieses Textverarbeitungssystems gibt, nichts mit den angebotenen Formularen anfangen.

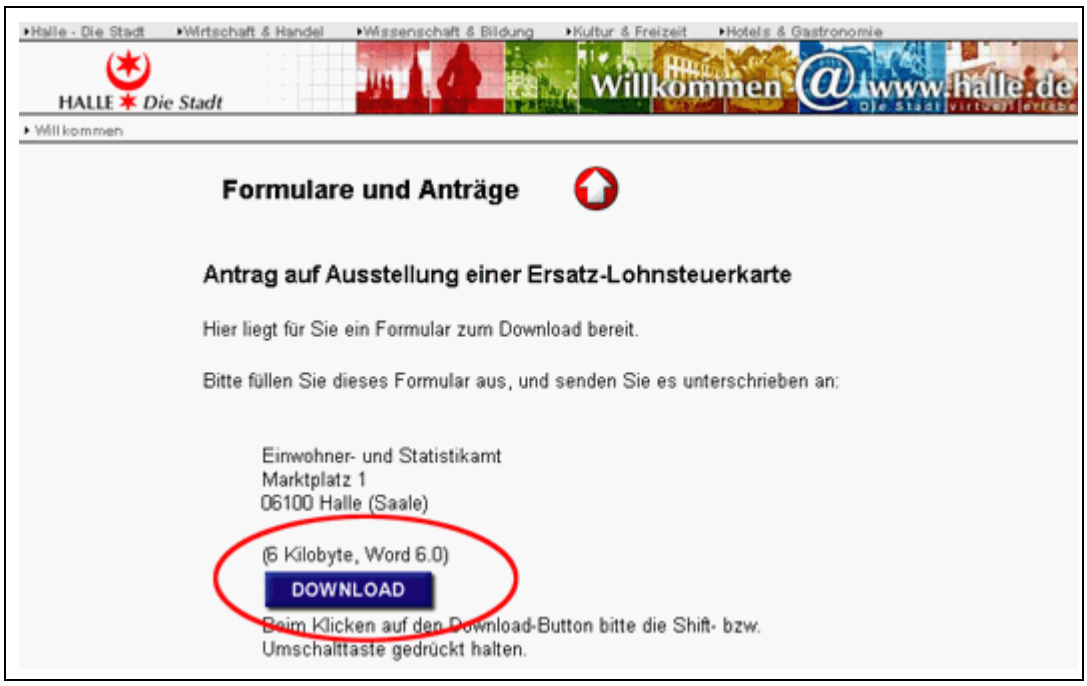

Abb. 1-2: Formulare zum Download (aus [HAL01])

Andere Kommunen (z.B. Beelitz [Bee01]) bieten dazu alternative Formate zum Download an, wie HTML oder *Portabel Document Format* (*PDF*). Beide Formate haben den Vorteil, dass sie bis auf exotische Ausnahmen von jedem Rechner angezeigt und verarbeitet werden können, ohne ein kommerzielles Programm installiert haben zu müssen. Das kostenlose Browser-Plugin zum Anzeigen von PDF-Dateien wird im Internet von der Firma Adobe [ADO01] angeboten. Das Angebot der Formulare in elektronischer Form stellt zwar schon eine Erleichterung für die

Bürger (ein Behördenbesuch entfällt) und eine Entlastung der Stadtverwaltung dar, jedoch ist dies nur als Anfang einer neuen Form der „Behördengänge“ anzusehen. Die bisher praktizierte Vorgehensweise (Abb. 1-3) weist einen erheblichen Nachteil auf. Das Formular, welches in elektronischer Form vorliegt, kann mit Hilfe des Computers elektronisch ausgefüllt werden. Danach muss es ausgedruckt an die Verwaltung geschickt werden (Medienbruch). Dort wird der Antrag aber wieder elektronisch verarbeitet.

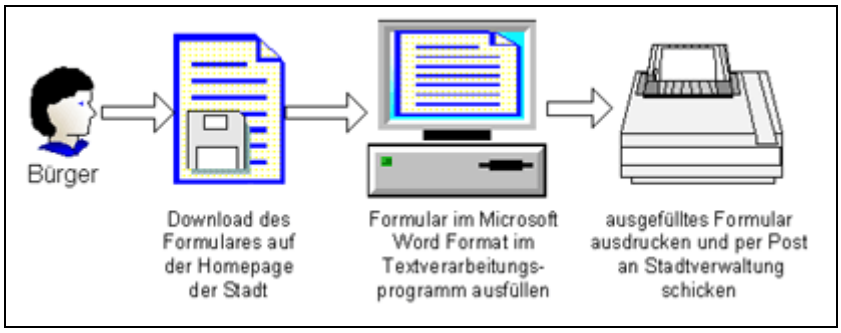

Abb. 1-3: Verlauf bisheriger Antragstellung (Bürgersicht)

Ein Grund für diese Vorgehensweise ist das Fehlen von Systemen, die in der Lage sind, über das Internet übermittelte Daten von Bürgern ordnungsgemäß zu speichern und den Mitarbeitern zur Bearbeitung anzubieten. Des Weiteren müssten sich die Mitarbeiter in den Verwaltungen an neue Wege und Verfahren gewöhnen, wozu nicht alle sofort bereit sein werden. Die Verarbeitung der ausgedruckten Formulare bedarf keinerlei Umstellung ihrer bisherigen Arbeitsweise. Würden die Daten auf elektronischem Wege in die Verwaltung kommen, so müssten neue Vorschriften geschaffen und Schulungen durchgeführt werden. Dies alles wirkt hemmend gegen jede Neuerung und Nutzung der neuen Möglichkeiten in großen Apparaten wie einer Stadtverwaltung.

Die ebenfalls in Halle mögliche Reservierung eines KFZ-Wunschkennzeichens über das Internet funktioniert schon etwas fortschrittlicher und stellt keine Anforderungen an irgendwelche zusätzliche Software auf dem Bürgerrechner. Hier kann man als Bürger in einem HTML-Formular im Browser seine Wunschkombination angeben. Dieser Wunsch wird dann per *elektronischer Post* (*E-Mail*) an einen zuständigen Mitarbeiter versandt. Somit muss der Bürger keinen Schritt vor die Tür setzen - bis er das Kennzeichen abholt.

Solche oder ähnliche Dienste werden von verschiedenen Kommunen angeboten (vgl.

[KON01]). Interaktive Dienste werden mit Sicherheit in der Zukunft höhere Bedeutung erlangen. Dies zeigen auch die Bemühungen des Bundes zur Einführung der digitalen Signatur, welche für viele online gestellte Anträge die grundlegende Voraussetzung darstellt.
Sollen An- und Verträge, für die eine rechtsverbindliche Unterschrift erforderlich ist, ohne die Nutzung der digitalen Signatur online angeboten werden, so ist dies meist mit einem umständlichen Verfahren verknüpft wie das Beispiel des Landkreises Cham zeigt (vgl. [CZ4801], S.12). Dort werden den Bürgern zur Zeit 180 verschiedene Formulare im PDF angeboten. Ein benötigtes Formular muss zuerst heruntergeladen, ausgedruckt und von Hand ausgefüllt werden. Danach können die fertigen Formulare wieder eingescannt und als Anhang zu einer E-Mail an die zuständige Dienststelle geschickt werden. Die passende E-Mail Adresse ist dabei auf der Download-Seite zu finden und muss später manuell im Mail-Programm eingegeben werden. Um diesen Dienst auch Nutzern zugänglich zu machen, die nicht über die dazu nötige Peripherie verfügen, wurden telematische Bürgerämter in den Gemeinden eingerichtet.
Der nächste Schritt hin zum vollständigen virtuellen Behördengang ist nun, und das ist auch gleichzeitig das Thema dieser Arbeit, Anfragen oder Anträge, die von den Bürgern ausgehen, direkt in elektronischer Form auf einem Server zu speichern und nicht mehr per E-Mail an einen einzelnen Ansprechpartner, meist den Webmaster, zu senden. Aufbauend darauf soll die weitere Bearbeitung der Anträge vom System gesteuert und protokolliert und damit effizienter und leichter nachvollziehbar werden. Die genauen Aufgaben des zu entwickelnden Systems werden in Abschnitt 1.4 erläutert.

1. 2 Vorteile gegenüber Offline-Antragstellung

Bisher werden viele Anträge, die per Post an die Stadtverwaltungen geschickt werden, von Mitarbeitern in maschinenlesbare Form gebracht. Dies geschieht meist durch Eingabe der Daten oder durch Scannen des Antrags. Dadurch dass die Antragsdaten bei der interaktiven Online-Antragstellung sofort in elektronischer Form bereitstehen, entfällt diese mühsame Eingabe, wodurch Arbeitskräfte eingespart werden können. Daneben erübrigt sich die Herstellung und der Druck von Papierformularen, da jeder Antragsteller seinen Antrag selber ausdruckt.
Gegenüber vielen bisherigen Lösungen (siehe Beispiel Halle) bietet eine rein web-basierte Antragsstellung (d.h. die Formulare können von einem Browser angezeigt werden) für den Bürger

einen klaren Vorteil. Er muss keine zusätzliche Software erwerben, um die Anträge, die von der Stadtverwaltung in einem bestimmten Format zum Download angeboten werden, anzusehen, auszufüllen und auszudrucken.
Durch Wegfall des Postweges, der bei einer normalen Antragstellung sowohl auf dem Weg zur Stadtverwaltung als auch auf dem Weg zurück entsteht, kommt es zu einem erheblichen Geschwindigkeitsvorteil. Leider muss man dazusagen, dass heutzutage die Mitteilung, ob der Antrag genehmigt wurde oder nicht, oftmals noch auf herkömmlichem Postweg zum Antragsteller übersandt wird. Das hängt mit der im Abschnitt Sicherheit (vgl. Abschnitt 3.6) näher geschilderten, noch nicht ausreichend verbreiteten digitalen Signatur zusammen. Sollte die digitale Signatur jedoch schneller als bisher weite Verbreitung finden, so ergeben sich damit noch größere Vorteile. Denn nicht nur der Postweg vom Amt zum Bürger würde auf diese Weise wegfallen, vielmehr könnten dann auch viele Anträge online gestellt werden, welche heute aufgrund der meist fehlenden Hilfsmittel für das digitale Pendant zur handschriftlichen Unterschrift nicht online gestellt werden dürfen.

1. 3 Nachteile gegenüber Offline-Antragstellung

Der wohl größte Nachteil, der sich beim Einsatz des Systems ergibt, ist der, dass Mitarbeiter für den Umgang mit dem neuen System geschult werden müssen. Wenn es in der Verwaltung bisher noch keinen Zugang zum Internet gibt, so kommt auch noch ein erheblicher finanzieller Aufwand hinzu, der durch die Anschaffung und Einrichtung des Systems entsteht.
Es gibt bisher in den Kommunen noch keine einheitliche Verfahrensweise zur Archivierung der Online-Anträge. Bisher werden die Antragsdaten in gedruckter Form „abgelegt“. Das ist natürlich noch nicht befriedigend, da dies wieder einen Medienbruch darstellt, der sich aber zunächst durch das Fehlen von verbindlichen Vorschriften nicht abstellen lässt. Es ist klar, dass bis zur vollständig elektronischen Antragsbearbeitung von der Eingabe bis zur Archivierung noch verschiedene Anstrengungen (vor allem vom Gesetzgeber) unternommen werden müssen, damit einheitliche Richtlinien (Rechts-)Sicherheit bei ausschließlich elektronischer Archivierung ermöglichen.

1. 4 Ziel- und Aufgabenstellung

Ziel dieser Arbeit ist die Konzeption eines Systems, welches die bisher übliche Art der Online-Antragstellung an Gemeinden (wie in Abschnitt 1.1 beschrieben) durch Einsatz moderner Techniken beschleunigt und vereinfacht. Aufgrund der häufig geringen Finanzmittel der Gemeinden soll das System hauptsächlich auf Geräten aufsetzen, die wenn möglich in der Verwaltung schon vorhanden oder mit geringem Aufwand zu beschaffen sind. Vor allem soll gezeigt werden, wie eine kostengünstige Lösung zu realisieren ist und welche Soft- und Hardwarekomponenten dazu nötig sind.

Die Aufgaben des zu entwickelnden Systems lassen sich wie folgt zusammenfassen: Die Daten, die durch Online-Antragstellung von Bürgern an die Stadtverwaltung anfallen, werden zentral auf einem Server gespeichert und den zuständigen Mitarbeitern zur Bearbeitung „zugestellt". Deren Bearbeitungsvorgänge werden protokolliert und der Bearbeitungsstatus der Daten wird gespeichert. Außerdem unterstützt das System die Mitarbeiter bei der Suche nach bestimmten Anträgen durch eine Rechercheschnittstelle und bietet auch eine leichte Erweiterbarkeit für zusätzliche Anträge. Selbst umfangreichere Aufgaben können damit komfortabel gelöst werden. Wenn zum Beispiel zu einem Antrag Genehmigungen aus mehreren Ämtern benötigt werden, soll die Verteilung der Daten an die entsprechenden Ämter durch das System erfolgen.

Aus informationstechnischer Sicht gibt es verschiedene Ansätze, die zur Lösung der oben beschriebenen Aufgabe eingesetzt werden könnten, zum Beispiel Workflow-Management oder Dokumenten-Management. Implementierungen dieser Ansätze führen zu sehr komplexen und damit teuren und nicht ohne umfangreiche Schulung einzusetzenden Systemen. Da dies wie oben erläutert vermieden werden soll, kommt die Einführung solcher Systeme nicht in Betracht. Es soll deshalb versucht werden ausgewählte Funktionalitäten dieser Ansätze wie Vorgangssteuerung, Speicherung und Präsentation mit einfachen Mitteln nachzubilden und eine mit geringem Aufwand umzusetzende Lösung zu finden, die damit klar von Workflow- oder Dokumenten-Management-Systemen abgegrenzt ist.

Im Folgenden wird zunächst ein Konzept für ein System entwickelt, welches die oben genannten Aufgaben erfüllt. Anschließend werden die genannten alternativen Ansätze Workflow-Management und Dokumenten-Management vorgestellt und vor allem hinsichtlich der Funktionalitäten, die im zu entwickelnden System eingesetzt werden sollen, untersucht (Kapitel

2). Kapitel 3 dient der Darstellung der technischen Grundlagen wie Kommunikationsprotokolle, Datenbanken und Sicherheit. Anschließend wird in Kapitel 4 die Implementierung des Systems beschrieben. Den Abschluss bilden in Kapitel 5 eine Zusammenfassung der Ergebnisse und ein Ausblick auf weitere Entwicklungen.

2 Analyse

2. 1 Komponenten des Gesamtsystems

Im Folgenden werden zunächst die Aufgaben des Systems herausgearbeitet und anschließend die wesentlichen Prozesse dargestellt:

1. Das System soll für die Mitarbeiter der Stadtverwaltung eine Unterstützung sein, um die Daten, welche bei der Stellung von Anträgen über das Internet anfallen, komfortabel in elektronischer Form abzulegen.

2. Gehen neue Anträge ein, soll eine Benachrichtigung an die jeweiligen Mitarbeiter geschickt und damit die Bearbeitung des Antrags angestoßen werden. Die einzelnen Bearbeitungsschritte, die die Sachbearbeiter mit den Anträgen ausführen, sollen protokolliert werden. Damit möchte man erreichen, dass genau festgestellt werden kann, wie weit ein Antrag in seiner Bearbeitung fortgeschritten ist und welche Mitarbeiter ihn bearbeitet haben.

3. Angestrebt wird weiterhin eine Komponente, die es ermöglicht, nach bestimmten Vorgaben Anträge im System zu finden und auszugeben.

4. Wichtig für die reibungslose Bearbeitung der eingegangenen Antragsdaten ist die Zuordnung der Mitarbeiter, zu bestimmten Anträgen. So sollen Mitarbeiter des Ordnungsamtes nur ihre Anträge bearbeiten müssen und nicht noch mit den Antragsdaten des Hoch- und Tiefbauamtes konfrontiert werden, wofür sie gar nicht zuständig sind. Hier ist eine Unterscheidung von Zuständigkeiten einzubauen. Dabei ist zu beachten, dass die Benachrichtigung über neue Daten nicht nur eine Person erhält, sondern alle Mitarbeiter, in deren Zuständigkeitsbereich der zugehörige Antrag fällt. Damit soll verhindert werden, dass der Antrag unbeachtet liegen bleibt, wenn ein Sachbearbeiter nicht im Hause ist (zum Beispiel wegen Krankheit), sondern durch andere Mitarbeiter bearbeitet werden kann.

5. Auch die Integration neuer Anträge und die Verwaltung der anfallenden Daten sollen durch das System erledigt werden. Dabei sollen einige Komponenten des Systems dem Bürger, welcher über das Internet auf das System zugreift, zugänglich gemacht werden, andere, welche zumeist aus datenschutzrechtlichen Gründen nicht öffentlich gemacht werden dürfen (man denke vor allem an die Daten von anderen Antragstellern), bleiben nur einigen Bearbeitern innerhalb der Stadtverwaltung vorbehalten. Für diese Abgrenzung und Überwachung wird eine Sicherungskomponente benötigt, welche dann gleichzeitig auch dem Schutz des internen Netzes in der Verwaltung (Intranet) dient.

Die Abbildung 2-1 zeigt die einzelnen Teilkomponenten des Gesamtsystems sowie ihr Zusammenspiel.

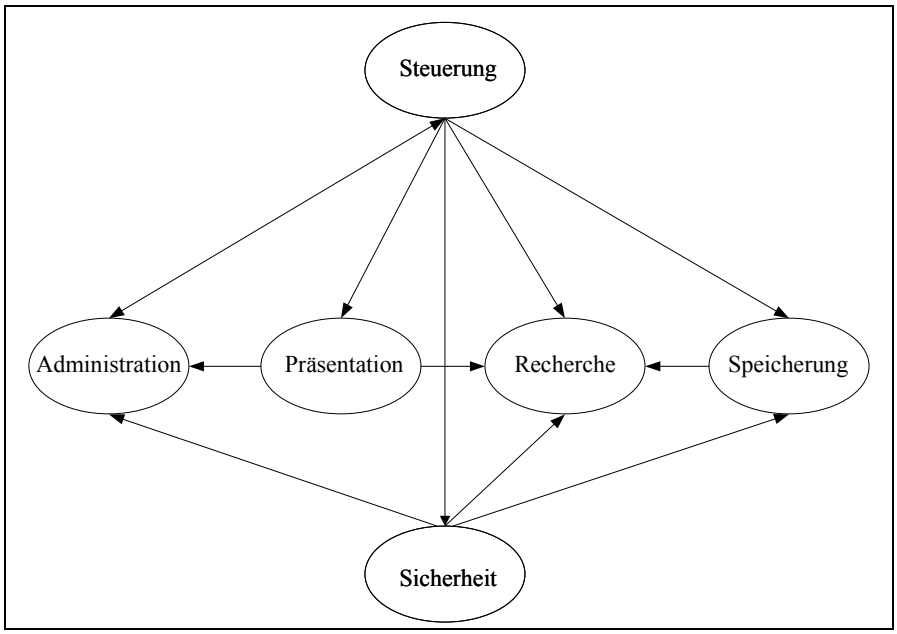

Abb. 2-1: Systemkomponenten

Die **Steuerung**skomponente ist dabei das Grundleitsystem. Sie bestimmt in welcher Weise eine Komponente zu einem bestimmten Zeitpunkt zum Einsatz kommt. Einige Eigenschaften der Steuerungskomponente lassen sich im Nachhinein verändern. Dies wird über die **Administration** möglich. Die **Recherche** bedient sich der gespeicherten Daten (**Speicherung**) und benutzt zur Anzeige, ebenso wie die Administration, das **Präsentation**smodul. Die Kontrolle über den berechtigten Zugriff auf sensible Teile des Systems sowie auf die Daten, die mitunter dem Datenschutzgesetz unterliegen, übernimmt die **Sicherheit**skomponente. Im Folgenden werden

diese Komponenten näher erläutert.

2. 1. 1 Die Steuerung

Die Steuerungskomponente kontrolliert den gesamten Betrieb des Systems. Sie entscheidet was geschieht und welche Aktivitäten ausgeführt werden, wenn neue Antragsdaten eintreffen oder wenn neue Anträge ins System eingepflegt werden sollen. Sie verwaltet die Zugangsberechtigungen aller beteiligten Personen (Mitarbeiter und Bürger) zu den verschiedenen Bereichen des Systems. Auch der Ort der Speicherung der Daten sowie zum Teil deren Sicherheit (Schutz vor unautorisiertem Zugriff) wird durch sie bestimmt.

2. 1. 2 Die Administration

Zur Administration des Systems sollte den dazu berechtigten Nutzern eine Oberfläche präsentiert werden, die auf einfache und leicht verständliche Art und Weise Einstellungen am System zur Änderung anzeigt. Über sie werden auch die Abläufe (beispielsweise zur Bearbeitung von Anträgen) definiert, die Nutzer den Anträgen zugeordnet oder neue Anträge eingefügt. Außerdem wird festgelegt, welche Bearbeiter für welche Anträge zuständig sind.

2. 1. 3 Die Präsentation

Wichtig für den Wiedererkennungswert ist eine einheitliche Oberfläche. Das hilft den Nutzern sich in einem System besser zurechtzufinden. Sowohl die Administration als auch die Recherche sollten deshalb ein einheitliches Format haben. Ungünstig wäre zum Beispiel im Extremfall, dass für die Administration eine Textdatei in einem Texteditor bearbeitet wird und zur Recherche ruft man ein Java-Programm auf, welches einige bunte Eingabefelder anbietet. Hier wird man erstens nicht erkennen, dass diese beiden Programme zu einem System gehören und zweitens findet man diese Fragmente schlecht, weil sie keine gemeinsame Wurzel haben. Besser wäre eine einheitliche Oberfläche, zum Beispiel ein Fenster, in welchem man bestimmte Funktionen auswählen kann, die sich dann im gleichen Stil in diesem Fenster bearbeiten lassen. Andererseits ist auch eine komplett über einen Browser zu bedienende Oberfläche denkbar. Dazu sind aber weitere Produkte notwendig, wie ein Web-Server und ein Übertragungsmedium. Diese Variante hätte aber den Vorteil, dass sie unabhängig von dem verwendeten Betriebssy-

stem funktioniert, da heutzutage für jeden Arbeitsplatzrechner ein passender Browser existiert.

2. 1. 4 Die Recherche

Durch die Recherche soll es den Mitarbeitern möglich gemacht werden, schnell und gezielt Daten aus gestellten Anträgen zu finden. Wenn die Daten in einzelnen Textdateien im Dateisystem abgespeichert werden, könnte man eine Suchfunktion wie sie im Microsoft Windows implementiert ist nutzen, um die gewünschten Daten ausfindig zu machen. Andererseits können die Daten auch in einer Datenbank strukturiert abgelegt sein. Hier kann man ein komfortableres Programm entwickeln, welches auch schneller und zuverlässiger die gewünschten Daten extrahiert. Dann sind auch andere Optionen denkbar, wie der Ausschluss von bestimmten Schlüsselwörtern oder die Eingrenzung auf bestimmte Attribute. Eine weitere Möglichkeit ist, dass auch die Antragsdaten aus schriftlichen Anträgen mit ausgewertet werden sollen. Dann muss natürlich auch ein Erkennungs- und Extraktionsmodul für die als Bilder abgespeicherten schriftlichen Anträge integriert sein, weil die Auswertung der Bitmaps nicht gerade trivial ist. Für diese Fälle gibt es auf dem Markt fertige Dokumenten-Management-Systeme, die genau diese Aufgaben lösen. In dieser Arbeit sollen aber nur die schon in elektronischer Form vorliegenden Daten der Bürger, welche über das Internet übermittelt werden, betrachtet werden. Deshalb entfällt eine zusätzliche Konvertierungskomponente.

2. 1. 5 Die Speicherung

Wie bereits angesprochen können die Daten entweder in Textdateien im Dateisystem oder in einer Datenbank gespeichert werden. Kommen noch Antragsdaten, die auf herkömmliche Weise schriftlich übermittelt wurden, hinzu, dann liegen diese (nach einer Übertragung) meist in einer computerlesbaren Form (meist Bitmaps) vor. Für eine Ablage im Dateisystem spricht die Einfachheit. Es werden keine zusätzlichen Produkte benötigt, wie es bei einer Datenbank der Fall wäre.

Wie auch in Dokumenten-Management-Systemen üblich (vgl. Abschnitt 2. 3. 2) ist es in jedem Fall notwendig, eine oder besser mehrere Kopien der Daten auf separaten Datenträgern zu halten, welche auch getrennt aufbewahrt werden sollten, um einen Totalverlust der Daten bei einem Problem mit dem Rechner, auf dem die Daten gespeichert werden, zu vermeiden. Sollen

später einmal Anträge mit der Digitalen Signatur eingereicht werden können, so sind bei der Archivierung der Daten besondere Vorgaben zu erfüllen. Auch wenn die Digitale Signatur bereits jetzt rechtsverbindlich ist, hat das Bundeswirtschaftsministerium erst im Juli 2001 das Projekt Archisig gestartet. Ziel dieses zwei Jahre dauernden Projektes ist, dass Industrie, Wissenschaft und Anwender sich über die beweiskräftige Langzeitarchivierung digital signierter Dokumente verständigen und konkrete Vorgaben erarbeiten. Bis dahin wird jedes System rechtlich anfechtbar sein, was Anbieter und Anwender vorerst abgeschreckt werden (vgl. [CZ4101]).

2. 1. 6 Die Sicherheit

In Abhängigkeit von der Art der Realisierung des Systems müssen unterschiedliche Vorsichtsmaßnahmen ergriffen werden, um die anfallenden Daten vor Unbefugten zu schützen. Dadurch dass die Anträge über das Internet gestellt werden, ist schon ganz am Anfang dafür zu sorgen, dass vertrauliche Daten nicht durch Fremde erspäht oder verfälscht werden können. Aber auch die Daten, welche der Steuerung des Systems dienen, müssen gegen Angreifer geschützt werden. Dies könnte durch eine völlige Entkopplung des Netzes, in welchem sich das System befindet, vom Internet erreicht werden. Aber auch „interne“, innerhalb der Stadtverwaltung gestartete Angriffe sollten unwirksam sein. Viele Firmen vertrauen deshalb auf ausgeklügelte Firewall-System, die an dieser Stelle zum Einsatz kommen können.

2. 2 Wesentliche Prozesse

Im folgenden Abschnitt werden die Hauptprozesse des Systems, der Weg der Daten vom Bürger zur Verwaltung und die Speicherung des Antrags sowie die Vorgangssteuerung zur Bearbeitung des Antrags in der Verwaltung vorgestellt.
Der Weg der Daten vom Browser des Bürgers zum Server der Verwaltung sieht folgendermaßen aus (vgl. Abbildung 2-2): Um die Dienste, die die Stadtverwaltung dem Bürger anbietet, nutzen zu können, muss dieser eine Verbindung zum Internet haben. Mittels eines Browsers kann er nun die Website seiner Stadt aufrufen und bis zu den (interaktiven) Anträgen durchblättern. Klickt er nun auf den Link zu einem Formular wird dieses in seinem Browser geöffnet. Sicherheitshalber sollte dieses Formular und dessen Auswertungsskript (an welches die Daten

dann gesendet werden) auf einem SSL-Server liegen (vergleiche Abschnitt 3. 5 Sicherheit). Dadurch wird gewährleistet, dass die übermittelten Daten weder von anderen mitgelesen noch von diesen verändert werden können. Der Bürger füllt nun das Formular in seinem Browser aus und schickt es mit seinen Daten an den Server der Stadtverwaltung zurück. Nachdem das Skript abgearbeitet wurde, welches die Daten des Bürgers empfangen, diese in die Speicherungskomponente aufgenommen und den Bürger über den Eingang der Daten informiert hat, startet die Vorgangssteuerung in der Behörde ihre Arbeit (vgl. Abb. 2-3).

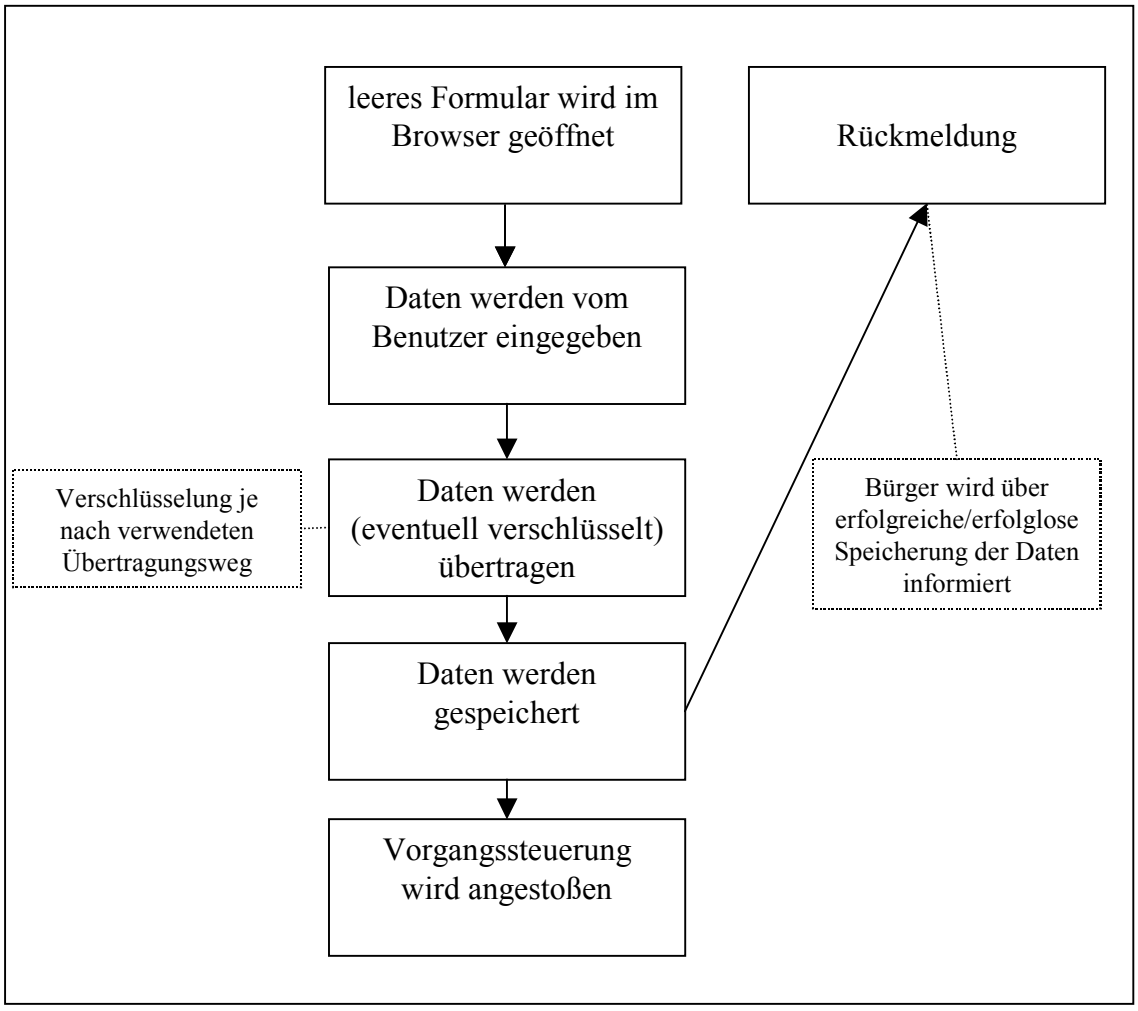

Abb. 2-2: Aktivitäten vor Start der Vorgangssteuerung

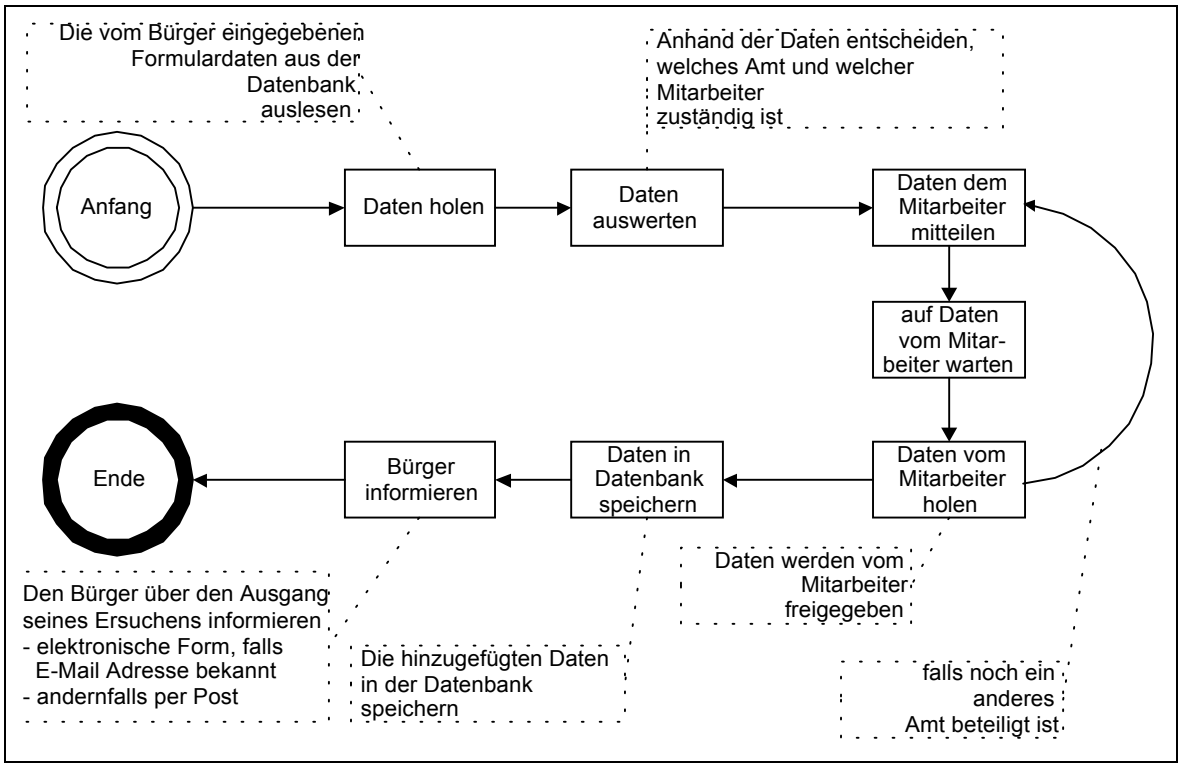

Abb. 2-3: Vorgangssteuerung

An dieser Stelle kann ein automatischer Abbruch des Vorgangs erfolgen, wenn bestimmte Voraussetzungen erfüllt sind, die ebenfalls in der Datenbank angegeben sein müssen. Zum Beispiel kann die Vorgangssteuerung den Antrag abweisen, wenn der Antragsteller innerhalb einer bestimmten Frist mehrere gleichlautende Anträge stellt, weil der erste vielleicht abgewiesen wurde und er es erneut versuchen möchte. Aber auch fehlerhafte Anträge, zum Beispiel falsche oder unvollständige Angaben, könnten zu einer Abweisung führen. Dies kann nun in einem sehr frühen Stadium erkannt werden, ohne dass sich erst ein Mitarbeiter mit dem Antrag auseinandersetzen muss. Das spart Zeit und somit auch Geld. Natürlich sollte eine solche automatische Abweisung nicht unbemerkt verlaufen. Es sollte zumindest der zuständige Sachbearbeiter und vor allem der Antragsteller selbst darüber informiert werden, damit dieser nicht glaubt, dass sein Antrag ordnungsgemäß eingegangen sei.

Bricht die Vorgangssteuerung nicht ab, erfolgt die Benachrichtigung der zuständigen Mitarbeiter über den Eingang der neuen Daten. Diese Benachrichtigung wird vorerst mit einer einfachen E-Mail realisiert, deren Inhalt angibt, dass ein Bürger einen Antrag (z.B. auf Baumfällung) gestellt hat, und die einen Link enthält, der den Mitarbeiter direkt zum neuen Datensatz führt. Die

Information, welche Mitarbeiter eine Benachrichtigung erhalten müssen, erhält die Steuerungskomponente aus den Antragsdaten, die sie mit den Zuständigkeiten in der Speicherungskomponente abgleicht.
Der Unterschied zu bisherigen Lösungen bei Anträgen über das Internet ist leicht erkennbar. Während bei den bisherigen Online-Anfragen oder -Anträgen die vom Bürger eingegebenen Daten per Mail an den Empfänger geleitet werden, kommt hier ein System zum Einsatz, welches nicht nur die Daten entgegennimmt, abspeichert und zuständige Mitarbeiter informiert, hier wird ein weitaus wichtigerer Schritt getan. Die Daten werden mit diesem Verfahren an einer zentralen Stelle (der Datenbank) hinterlegt und sind nun für weitere Recherchen von jedem authentifizierten Nutzer (berechtigte Sachbearbeiter der Stadtverwaltung) leicht einsehbar.
Dies ist erheblich komfortabler als die Variante, die eine E-Mail vom Bürger zur Stadtverwaltung als Kommunikationsgrundlage hat. Da dort die Daten nur an eine Person geschickt werden, kann es nicht ohne weiteres festgestellt werden, wenn diese Person, aus welchen Gründen auch immer (vielleicht Krankheit), es versäumt die Daten weiterzuverarbeiten. In dem in der vorliegenden Arbeit zu entwickelnden System ist es möglich eine automatische Überprüfung zu integrieren, welche von Zeit zu Zeit die eingegangenen Anträge auf ihre Bearbeitungszustände hin untersucht. Versäumnisse können hier nun schnell erkannt und behoben werden.
Dadurch dass die Anträge an einer zentralen Stelle gespeichert sind, können alle Mitarbeiter, die Zugriff auf diese Daten haben, umfangreiche Dienste nutzen. So können zum Beispiel statistische Auswertungen Auskunft darüber geben, welche Angebote sehr häufig genutzt werden und wo es damit zu Engpässen bei der Bearbeitung der Anträge kommen könnte. Auch die Suche nach bestimmten Daten ist nun leicht durchführbar. Bei diesen vielen Vorteilen ist es natürlich auch wichtig, dass es durch die zentrale Haltung der Daten auch ein großes Sicherheitsbedürfnis gibt. Dies wird in Abschnitt 3. 5 näher beleuchtet.

2. 3 Alternative Ansätze

2. 3. 1 Workflow-Management

Zuerst einmal soll der Begriff Workflow-Management unter Zuhilfenahme der Definitionen der *Workflow Management Coalition* (*WfMC*), einer Vereinigung von Herstellern und Forschungs-

einrichtungen, die ein Workflow-Referenzmodell entwickelt hat, geklärt werden.
Arbeitsabläufe des täglichen Lebens (wie zum Beispiel Bestellungen bei Lieferanten) können als eine Menge zusammengehöriger Aktivitäten, dem Anwendungsprozess, beschrieben werden. Dazu zählen nicht nur die automatisierten Aktivitäten, die durch Applikationen ausgeführt werden, sondern auch die manuellen, welche durch Nutzer durchgeführt werden.
Ist ein Prozess stark strukturiert und damit die Reihenfolge der einzelnen Aktivitäten ziemlich genau festgelegt, kann man eine Prozessvorlage, eine sogenannte *Workflow-Definition,* erzeugen. Diese besteht aus (vgl. [WfMC99], S. 7-17):

- einem *Kontrollfluss*, welcher die Aktivitäten des Anwendungsprozesses und deren Beziehungen untereinander beschreibt,
- dem *Datenfluss*, der den Weg von Daten (wie zum Beispiel Dokumenten) und Informationen vorgibt und
- zusätzlichen Informationen zu den einzelnen Aktivitäten (zum Beispiel Eigenschaften einer Aktivität).

Die Trennung und grafische Darstellung von Kontroll- und Datenfluss soll der Übersichtlichkeit dienen und der Fehleranfälligkeit entgegenwirken. Wird zur Veranschaulichung ein Graphmodell mit Knoten und Kanten genutzt, können die Aktivitäten der Workflow-Definition den Knoten und Beziehungen zwischen Aktivitäten den Kanten zugeordnet werden. Es sind aber auch andere Darstellungen möglich zum Beispiel Petri-Netze (vgl. [BAU96]) oder State- und Activity-Charts (vgl. [MWG+99]).
Um einen Arbeitsablauf letztendlich zu realisieren, wird von der jeweiligen Workflow-Definition eine *Instanz* erzeugt und ausgeführt. Dann kann die Abarbeitung der Aktivitäten des Kontrollflusses beginnen. Dabei werden alle benötigten Informationen wie Daten bereitgestellt, gegebenenfalls bestimmte Applikationen gestartet (automatische Aktivitäten) und Benachrichtigungen an beteiligte Nutzer geschickt (manuelle Aktivitäten).
Die Zuordnung der Nutzer zu den Aktivitäten erfolgt über das sogenannte *Rollen-* oder *Organisationsmodell*. Dabei werden bestimmte Rollen innerhalb der Verwaltung festgelegt wie zum Beispiel „Sachbearbeiter für Antrag xy“, denen die Nutzer zugeordnet werden. Jeder auszuführenden Aktivität werden eine oder mehrere Rollen zugeordnet. Ein Nutzer, der die entsprechende Rolle innehat, darf die Aktivität ausführen. Ein solches Rollenmodell wird benötigt um

Aufgabe vier aus Abschnitt 2.1, die Regelung der Zuständigkeiten der Mitarbeiter, zu erreichen. Das System, welches die Modellierung von Arbeitsabläufen in Workflow-Definitionen mit Kontroll- und Datenfluss ermöglicht und verschiedene Workflow-Definitionen und -Instanzen verwaltet, nennt man *Workflow-Management-System* (vgl. [WfMC99], S. 7-17). Abbildung 2-4 stellt die bisher verwendeten Begriffe in ihrem Zusammenhang grafisch dar, wobei aus Übersichtlichkeitsgründen auf die Integration des Rollenmodells verzichtet wurde.
Ein Workflow-Management-System besteht im Wesentlichen aus zwei Teilen (vgl. Abb. 2-5). Einem *Workflow-Editor*, welcher der Definition des Workflows dient, und einer sogenannten *Workflow-Engine*, welche die Erstellung und Abarbeitung der einzelnen Workflow-Instanzen ermöglicht und die Instanzen verwaltet. Die Workflow-Engine ist zudem verantwortlich für den Start von Applikationen bei automatischen Aktivitäten oder die Benachrichtigung von Personen bei manuellen oder halbautomatischen Aktivitäten.

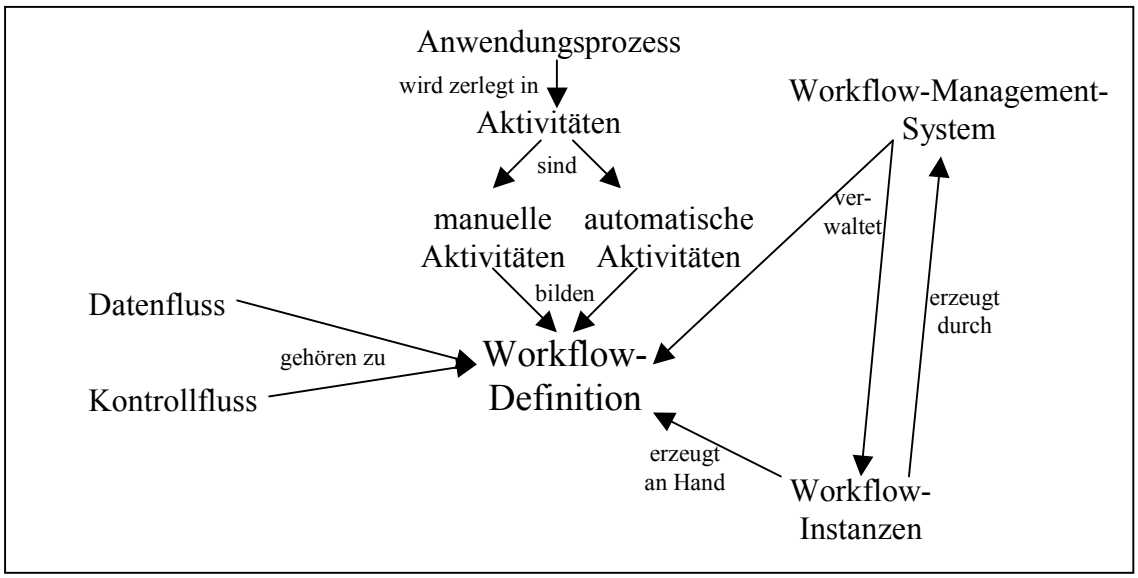

Abb. 2-4: Workflow - Überblick

Es ergeben sich zwei Zustände: einmal die Definition und Vorbereitung des Workflows unterstützt durch den Workflow-Editor - die *Buildtime* - und zum anderen die Abarbeitung einer Workflow-Instanz durch die Workflow-Engine - die *Runtime*.

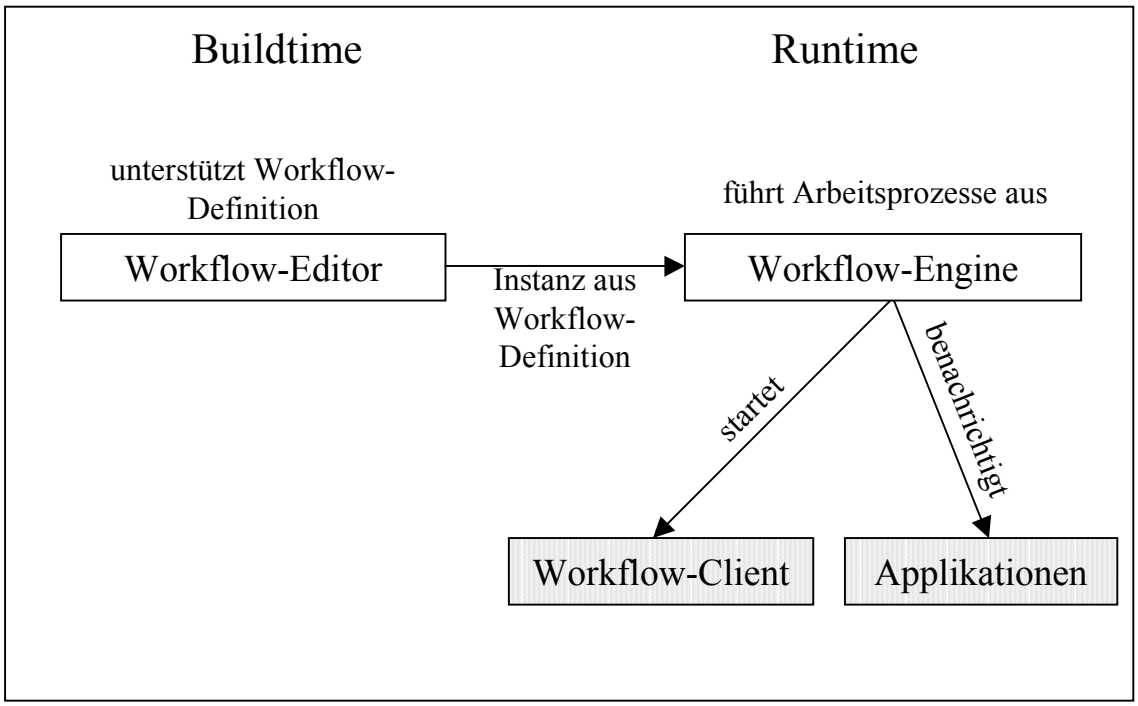

Abb. 2-5: Architektur eines Workflow-Management-Systems (nach [WfMC99])

Mit Hilfe eines solchen Systems ist es möglich jede erdenkliche Art von sich nicht häufig ändernden Prozessen abzubilden. Die Funktionalität der Workflow-Engine zur Kontrolle von Prozessen ist auch im hier zu entwickelnden System erforderlich um Aufgabe zwei aus Abschnitt 2.1, die Steuerung und Protokollierung der Antragsbearbeitung, zu erfüllen. Allerdings gibt es einige Gründe, warum ein solches System nicht für die Lösung der Problemstellung dieser Arbeit zum Einsatz kommt, die im Folgenden dargestellt werden.

Es gibt in der Verwaltung zur Bearbeitung der Anträge nur einen Prozess mit wenigen genau definierten Aktivitäten (vgl. Abschnitt 2.2), die untereinander noch recht übersichtliche Beziehungen haben, die sich auch einfach darstellen lassen. Die Wahrscheinlichkeit von fehlerhaften Prozessdefinitionen aufgrund hoher Komplexität ist also gering und damit eine Modellierungsunterstützung wie sie ein Workflow-Editor bietet nicht nötig. Da es nur Instanzen zu einer, nicht sehr komplexen Prozessdefinition gibt und voraussichtlich nicht hunderte von Instanzen gleichzeitig ausgeführt werden, erscheint auch der Einsatz einer Workflow-Engine zur Kontrolle und Verwaltung der Prozessinstanzen nicht sinnvoll, da diese ihre Stärken wie zentrale Ressourcenverwaltung, Kontrolle der Ausführung oder Bereitstellung der Daten erst ausspielen kann, wenn viele Instanzen von unterschiedlichen Prozessdefinitionen gleichzeitig ausgeführt werden.

Desweiteren soll bei der Problemlösung auf bereits vorhandene Komponenten innerhalb einer Stadtverwaltung aufgebaut werden. Da in vielen Ämtern ein Workflow-System erst für viel Geld gekauft, installiert und auch Personal geschult werden müsste, während Datenbanksysteme, Arbeitsplatzrechner und Webserver meist schon vorhanden und den Mitarbeitern vertraut sind, wird das zu entwickelnde System auf diesen Komponenten aufbauen und kein Workflow-System integrieren.

2. 3. 2 Dokumenten-Management

Wenn man statt den Aktivitäten die Daten in den Mittelpunkt seiner Betrachtung stellt, kommt man schnell zu einem Dokumenten-Management-System. Von einem Dokumenten-Management-System ist dann die Rede, wenn zur Verwaltung und Vorgangsbearbeitung zusätzlich der gesamte Lebenszyklus von Dokumenten betrachtet werden soll. Dabei werden unter dem Begriff Dokument neben Textdateien oder digitalen Bildern und Videos alle Objekte der *elektronischen Datenverarbeitung* (*EDV*) verstanden. Ein solches System dient der Erzeugung, Weiterleitung, Bereitstellung und Archivierung dieser Objekte. Damit sind Teilkomponenten eines solchen Systems für die Realisierung der Aufgaben eins, Speicherung der vom Bürger eingegebenen Daten, und drei, Recherche auf den gespeicherten Anträgen, aus Abschnitt 2.1 in Betracht zu ziehen.

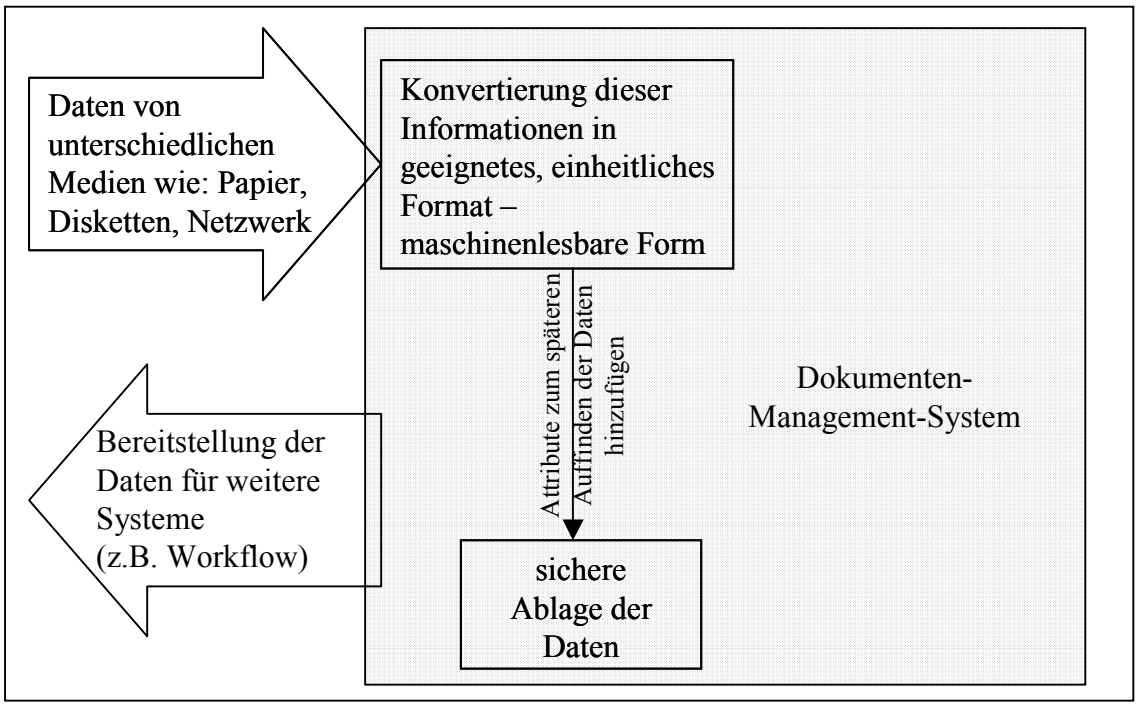

Abb. 2-6: Überblick Dokumenten-Management-System

Eine weitere wesentliche Aufgabe besteht in der Vermeidung von Medienbrüchen (vgl. [GSS99]). Diese würden zum Beispiel bei der einfachsten Lösung des Problems dieser Arbeit auftreten, wenn nämlich die von den Bürgern über das Internet übermittelten Daten auf ein anderes Medium übertragen werden, zum Beispiel durch Ausdrucken der Antragsdaten auf Papier. Das hätte zwar den Vorteil, dass an der bisherigen Antragsbearbeitung in der Verwaltung ab da keine Änderung notwendig wäre, da diese (in der Stadtverwaltung) ausgedruckten Anträge wie bisher weiterverarbeitet werden könnten. Da die Daten aber schon einmal (von den Bürgern) in elektronischer Form bereitgestellt wurden, erscheint diese Vorgehensweise objektiv gesehen als sehr ungeeignet zumal die Weiterverarbeitung der Daten in elektronischer Form erfolgt, das heißt die gerade ausgedruckten Antragsdaten müssten wieder in elektronisch lesbare Form gebracht werden. Die beste Lösung würde demnach die maschinenlesbaren Daten, welche von den Bürgern über das Internet übermittelt werden, entweder direkt als Eingabeparameter in vorhandene Bearbeitungssysteme übergeben oder ein eigenes Bearbeitungsprogramm zur Verfügung stellen, mit dessen Hilfe die Verarbeitung der Daten erfolgt. Die Vermeidung von Medienbrüchen (eine Teilfunktionalität eines Dokumenten-Management-Systems) ist also wie oben dargelegt ein Grundgedanke des in dieser Arbeit zu konzipierenden Systems.

Es gehört außerdem zur Aufgabe eines Dokumenten-Management-Systems die in elektronischer Form vorliegenden Daten auf geeigneten Speichermedien sicher abzulegen. Diese Teilfunktionalität dient vor allem der Erfüllung von Aufgabe eins (Speicherung der vom Bürger eingegebenen Daten) aus Abschnitt 2.1.
Das Dokumenten-Management-System bildet heute meist zusammen mit weiteren Komponenten, wie Groupware-Systemen oder Workflow-Systemen aber auch Intranet- und Internet-Diensten den Gesamtkontext einer elektronischen Informationsverarbeitung. Da in den meisten Verwaltungen Dokumenten-Management-Systeme wie auch Workflow-Systeme nicht vorhanden sind, sondern erst mit nicht unerheblichem finanziellen und personellen Aufwand eingeführt werden müssten und das gewünschte System auf vorhandenen Komponenten aufbauen soll, kommt der Einsatz eines Dokumenten-Management-Systeme zur Realisierung des gewünschten Systems nicht in Betracht.

2. 4 Voraussetzungen

2. 4. 1 Hardware

Folgende Hardware, die in Tabelle 2-1 zusammengestellt und weiter unten näher erläutert ist, wird von den verschiedenen Komponenten benötigt.

Punkt	benötigte Hardware
Sachbearbeiter	pro Sachbearbeiter: - Computer mit Netzwerkkarte und Verbindung zum internen Netz oder Lesegerät, wenn die Daten von einem anderem Medium als dem Netzwerk gelesen werden sollen
zentraler Speicher der Daten	- ein Server mit Verbindung zum Intranet und zum Internet - je nach eingesetzten Sicherheitskomponenten Verbindung zu Firewalls - Backup-System
Präsentation und Steuerung der Daten	ein bis zwei (Web-) Server mit Verbindung zum Internet und zum internen Netz

Tabelle 2-1: Benötigte Hardware-Komponenten

Die Sachbearbeiter in der Verwaltung müssen mit ihrem *Arbeitsplatzrechner* (*PC*) in der Lage sein, eine Verbindung zur Speicherungskomponente aufzubauen. Im Allgemeinen wird dies eine Netzwerkverbindung sein, über die die Daten von einem entfernten Server zum Mitarbeiter-PC übertragen werden. Hierbei muss im Rechner eine Kommunikationskomponente wie ein Modem oder eine Netzwerkkarte installiert und einsatzbereit sein.
Wird der schon vorhandene Webserver (welcher für die Bereitstellung der Anträge im Internet verantwortlich ist) auch für die Bearbeitung der Daten benutzt, so ist zu berücksichtigen, dass dieser Server neben den Zugriffen der Bürger über das Internet auch die Zugriffe aus der Verwaltung, die bei der Bearbeitung der Anträge und der Recherche anfallen, zu verarbeiten hat. Dies setzt eine hohe Übertragungsrate des Netzwerks (zumindest innerhalb der Verwaltung) voraus.
Um die Sicherheit zu erhöhen, sollte man einen weiteren WWW-Server einsetzen. Damit wäre es möglich, einen WWW-Server nur für die Zugriffe aus dem Internet bereitzustellen, welcher keine Skripte zur Verwaltung, Administration und Bearbeitung der Anträge beinhaltet. Diese Skripte würde man auf dem zweiten WWW-Server ablegen und zur Ausführung bringen. Die Sicherheitsrichtlinie wäre dann klar: Nur zu dem öffentlichen WWW-Server mit den allernötigsten Skripten darf von außerhalb des Intranets eine Verbindung aufgebaut werden.
Für den Fall, dass ein *Datenbank-Management-Systems* (*DBMS*) eingesetzt werden soll, muss der Datenbank-Server, welcher mit dem Webserver zusammenfallen kann (nur zu empfehlen, wenn das Budget nicht mehr hergibt), die Mindestanforderungen des verwendeten DBMS erfüllen. Reicht für eine MySQL Datenbank ein durchschnittlicher PC aus (vgl. [TCX01]), so sollte es bei der Verwendung von Oracle 9i schon besser ein doppelt so schneller Rechner mit dreimal soviel Speicherkapazität sein.
Weiterhin wird eine tägliche Sicherung der Datenbank mit geeigneten Methoden empfohlen. Dies kann aber durchaus auch nur ein Export des Datenbankinhalts in eine Datei und deren Speicherung auf *Compact Disc Recordable* (*CD-R*) umfassen.

2. 4. 2 Software

Die auf den jeweiligen Systemen benötigten Softwarekomponenten sind in der folgenden Tabelle zusammengefasst und werden danach näher erläutert:

System	Software
Sachbearbeiter PC	- E-Mail Programm - Webbrowser
Datenbank-Server	- Datenbank-Management-System - Backuptool
Webserver	- Webserver mit PHP Erweiterung - Mailserver

Tabelle 2-2: Benötigte Software-Komponenten

Für die Bedienung eines Java-Clients werden auf den Arbeitsplatzrechnern keine zusätzlichen Software-Komponenten benötigt, es sei denn der Client läuft als Applet innerhalb des Browsers. Dann muss ein solcher vorhanden sein.

Browser, E-Mail, Webserver

Bei einer HTML-basierten Version des Systems muss auf den Arbeitsplatzrechnern in den Ämtern ein Browser (zum Beispiel Netscape Communicator oder Microsoft Internet Explorer) ab der Version 4 installiert sein. Ein E-Mail Kommunikationsprogramm (zum Beispiel Qualcomm's Eudora, vgl. [EUD01]) ist für eine automatische E-Mail-Benachrichtigung beim Eingang neuer Antragsdaten nötig. Diese Komponenten sind für die gängigsten Betriebssysteme verfügbar. Somit hat man bei der Wahl desselben freie Hand. Da das in der Verwaltung auf den meisten PCs eingesetzte Microsoft Windows 2000 oder andere Mitglieder der Microsoft Windows Familie alle nötigen Anforderungen erfüllen, gibt es keinen Grund diese für die Mitarbeiter vertraute Arbeitsumgebung zu ändern.

Bei der HTML-basierten Version muss neben dem schon vorhandenen Webserver für die Bürger ein zusätzlicher für die Mitarbeiter verfügbar gemacht werden. Dieser dient dann zusammen mit dem Browser des Mitarbeiters als Präsentations- und indirekt, nämlich durch die Abarbeitung der Skripte, als Steuerungskomponente. Verbreitete Produkte sind z.B. der Apache Webserver oder der Microsoft Internet Information Server (vgl. auch [NC01]). Bei beiden Servern können zusätzliche Web-Server-Erweiterungen, die für die Verbindung zur Speicherungskomponente nötig sind, genutzt werden.

Datenbank-System

Als Datenbank-Management-Software können alle gängigen Produkte eingesetzt werden. Einschränkungen kann man aus den Dokumentationen der verwendeten Erweiterung entnehmen. Beispiele für verwendbare Produkte sind: MySQL, Microsoft SQL Server, PostgreSQL, dBase, Informix, Interbase und Oracle. In Anbetracht der Tatsache, dass die Mittel der Kommunen sehr begrenzt sind, ist es vorteilhaft bei der Wahl des Datenbank-Management-Systems genau zu analysieren, welche Funktionalität und welchen Support-Umfang man unbedingt benötigt. Die Preisspanne für ein solches Produkt ist nämlich immens.

Während der Einsatz von MySQL für die Stadtverwaltung keinen Pfennig kostet, sind für den Einsatz von Oracle 9i mehrere tausend Euro fällig. So kostet laut Aussagen von Oracle [ORA01] eine zeitlich unbegrenzte, Ein-Prozessor-Version der Oracle 9i Enterprise Edition 40.000 US$. Da sollte man meinen, man hat leichte Wahl. Doch der Leistungsumfang beider Produkte ist sehr unterschiedlich. MySQL gehört zur sogenannten Open Source Software, das heißt der Quellcode ist frei verfügbar und wird von verschiedenen Leuten, die nicht notwendigerweise zu ein und demselben Konzern gehören, weiterentwickelt. Auch werden nicht alle Funktionen, die man von verschiedenen Produkten kennt, von MySQL in den bisher veröffentlichten Versionen unterstützt. So sucht man bisher Transaktionen und Trigger vergeblich (geplant ab Version 4.1, vgl. [TCX01]).

Oracle bietet mit seiner Lösung dagegen ein selbst entwickeltes Produkt an, welches sehr umfangreiche und ausgereifte Funktionen bietet. Ebenso umfangreich wie die Software selbst ist der Support. So kann man die komplette Installation und Optimierung durch Oracle vornehmen lassen und auch bei eventuellen Fehlern oder Fragen hat man einen Ansprechpartner. Dies alles führt zu diesem gewaltigen Preisunterschied.

Letztendlich hängt die Wahl des Produkts davon ab, ob das KnowHow zum Beispiel in den Reihen der eigenen Administratoren vorhanden ist oder ob man es gegebenenfalls teuer erkaufen muss. Erfahrungen aus Projekten zeigen jedoch deutlich, dass es in den Verwaltungen an solch qualifiziertem Fachpersonal mangelt. So werden für eine Verwaltung, welche keine eigenen Erfahrungen besitzt und auch nicht in der Lage ist, diese Erfahrungen billig durch einen Wartungsvertrag mit einer Firma zu bekommen, nur die teureren Produkte mit Support in Frage kommen.

2. 4. 3 Personal

Durch die Einführung dieses Systems werden verschiedene Spezialisten benötigt, welche entweder eingestellt oder aber von Firmen für die jeweils anstehende Aufgabe bestellt und bezahlt werden müssen. Dies betrifft vor allem die Konzeption und Installation des Firewall-Systems und die Installation der WWW- und Datenbank-Server. Mitunter sind solche Fachkräfte sogar schon in der Verwaltung tätig, so dass diese Aufgaben ohne Einkauf von Fachwissen erledigt werden können.
Für die Benutzung des Systems benötigt man keine zusätzlichen Arbeitskräfte. Diese Aufgabe übernehmen die Mitarbeiter, welche bis jetzt auch die Anträge bearbeitet haben.
Zur Administration, vor allem zum Entwurf von neuen Formularen wird eine Arbeitskraft mit HTML- und Internet-Kenntnissen benötigt. Dieses Wissen kann über eine kurze Schulung für einen Mitarbeiter erworben werden.
Zur Personalentlastung kommt es hingegen dadurch, dass die bisher in nicht maschinenlesbarer Form gestellten Anträge nicht mehr mühsam in maschinenlesbare Form übertragen werden müssen, da sie ab der Eingabe durch den Bürger in den Browser digital vorliegen.

2. 4. 4 Zusammenfassung

Abschließend wird zusammengefasst, aus welchen Hard- und Software-Komponenten das komplette Online-System besteht (vgl. Tabelle 2-3). Dies sind im einzelnen: der WWW-Server, der Datenbank-Server, die Firewall, das Internet, über welches der Bürger eine Verbindung zum WWW-Server der Stadtverwaltung aufbaut, das Intranet in der Verwaltung, welches den Verbindungsaufbau von den Mitarbeitern zum WWW-Server ermöglicht, die Arbeitsplatzrechner, von denen die Mitarbeiter das System bedienen und administrieren, sowie das Netzwerk welches all diese Komponenten verbindet.

Komponente	Funktion
Internet	Bietet die Möglichkeit, eine Verbindung zum WWW-Server der Stadtverwaltung aufzubauen
WWW-Server	- Anbieten von Informationen, die von der Stadtverwaltung hinterlegt wurden - Abarbeitung der Skripte, welche die vom Bürger übermittelten Daten der Speicherungskomponente übergibt - Auslesen der dynamischen Inhalte aus der Datenbank und deren Ausgabe in HTML - Interpretation der hinterlegten Skripte - Zugangskontrolle
Datenbank-Server	- speichert dynamische Inhalte, die vom WWW-Server ausgegeben werden (Ansprechpartner, Öffnungszeiten) - Speicherung der Daten von online gestellten Anträgen - Speicherung der Daten, die zur Steuerung des Systems notwendig sind
Firewall	dient der Verbindungskontrolle und zur Absicherung des internen Netzes (Intranet) sowie des WWW-Servers und des Datenbank-Servers vor unbefugtem Zugriff
Intranet	bietet die Möglichkeit, dass Verwaltungsmitarbeiter eine Verbindung zum Datenbank-Server und zum WWW-Server aufbauen können
Arbeitsplatzrechner	Grundlage zur Bearbeitung, Nutzung und Administration des Systems

Tabelle 2-3: Hard- und Software-Komponenten des Systems

Tabelle 2-4 gibt noch einmal einen Überblick über die benötigten Software-Komponenten und ihre Eigenschaften und listet Produktbeispiele auf. Die Funktionsweise der Produkte wird in Kapitel 3 näher erläutert.

Komponente	besondere Eigenschaften	Produktbeispiele
Präsentation (Browser)	Unterstützung von *Cascading Style Sheets* (*CSS*)	Microsoft Internet Explorer 5.5, Netscape Communicator 4.72
E-Mail-Programm		Qualcomm Eudora, Microsoft Outlook, Netscape Messenger
Webserver	Unterstützung von Erweiterungen wie PHP, JSP oder ASP (vgl. Abschnitt 3. 2. 2)	Microsoft Internet Information Server, Apache Webserver, IPlanet Web Server Enterprise Edition
Speicherung (DBMS)		MySQL, Oracle 9i, Microsoft SQL Server
Sicherungskomponente		Firewall 1
Mail		sendmail

Tabelle 2-4: Software-Komponenten und Produktbeispiele

3 Grundlagen

3. 1 Übertragungsprotokolle

3. 1. 1 Definition

Ein Protokoll beinhaltet Vorschriften und Formate zum erfolgreichen Datenaustausch zwischen Kommunikationspartnern. Dabei müssen beide Parteien das gleiche Protokoll nutzen, damit sie sich verständigen können. Ist dies nicht der Fall, so bricht der Informationsfluss ab oder es kommt erst gar keiner zustande. Im Gegensatz zu den Protokollen, die von einzelnen Herstellern entwickelt und nur für das Zusammenspiel von Produkten ein- und desselben Herstellers konzipiert wurden, benötigt man im heterogenen Internet herstellerübergreifende Protokolle. Einige wichtige werden in den nächsten Abschnitten vorgestellt.

3. 1. 2 TCP/IP

Das *Transmission Control Protocol / Internet Protocol* (*TCP/IP*) ist die grundlegende Protokollfamilie für das Internet. Daneben wird es in vielen *Local Area Networks* (*LANs*) eingesetzt. Es handelt sich dabei um den Nachfolger des *Network Control Protocols* (*NCP*). TCP/IP wurde von der InterNetwork Working Group entwickelt und ist seit 1983 im Einsatz. TCP/IP wird meist zur Bezeichnung für die gesamte Protokollfamilie verwendet, die ursprünglich für das US-Verteidigungsministerium entwickelt wurde, um heterogene Netzwerke zu verbinden.
Das IP ist für die Wegwahl (Routing) zuständig. Es arbeitet verbindungslos und paketorientiert, wobei aber keine gesicherte Datagrammübergabe geboten wird. Im Gegensatz zu IP arbeitet TCP verbindungsorientiert und implementiert eine sichere Prozess-Prozess-Kommunikation mit integriertem Multiplexing und Flusskontrolle.

3. 1. 3 HTTP

Ein Protokoll der Verarbeitungsschicht ist das *Hyper Text Transfer Protocol* (*HTTP*). Es dient der Kommunikation zwischen WWW-Client (Browser) und WWW-Server. HTTP ist textbasiert, das heißt es werden *ASCII* (*American Standard Code for Information Interchange*)-Zeichen für die Kommunikation genutzt. Dadurch ist eine Herstellerunabhängigkeit gegeben. Aktuelle HTTP-Version ist 1.1.
Zwei Komponenten machen HTTP aus: Anfragen von Browsern an den Server und auf der anderen Seite dessen Antworten an den Browser.
In HTTP sind folgende Anfragemethoden integriert (vgl. [Tan98], S. 727):

GET:	Anfrage zum Lesen eines Web-Objekts
HEAD:	Anfrage zum Lesen des Headers eines Web-Objekts
PUT:	Anfrage zum Speichern eines Web-Objekts
POST:	Anhängen einer benannten Ressource
DELETE:	Entfernen eines Web-Objekts
LINK:	verbindet zwei vorhanden Ressourcen
UNLINK:	löst eine vorhandene Verbindung zwischen zwei Ressourcen

Für diese Arbeit interessant sind vor allem die Methoden GET und POST, die in Abschnitt 3.5.1 noch genauer erläutert werden.

3. 1. 4 HTTPS

HTTP Secure (*HTTPS*) besteht aus dem HTTP mit dem Unterschied, dass für die Verbindung *Secure Socket Layer* (*SSL*) zum Einsatz kommt. Damit ist gewährleistet, dass die übertragenen (meist vertraulichen) Daten nicht von Unbefugten eingesehen werden können, da diese verschlüsselt übertragen werden. Nach den Vorgaben der *Internet Assigned Numbers Authority* (*IANA*) benutzt HTTPS standardmäßig Port 443. Die gängigsten Browser (in den aktuellen Versionen) unterstützen HTTPS, für einige andere gibt es SSL-Patches. Wenn man über den Browser eine gesicherte Verbindung aufbauen möchte, muss der *Uniform Resource Locator* (*URL*) statt http:// https:// als Protokollangabe beinhalten. Dann wird standardmäßig versucht eine Verbindung zum Port 443 des Servers herzustellen. Natürlich muss der Webserver dafür konfigu-

riert worden sein.

3. 1. 5 SMTP

Das *Simple Mail Transfer Protocol* (*SMTP*) ist ein Protokoll der Verarbeitungsschicht. Es bildet die Grundlage für den Austausch von *Electronic Mail* (*E-Mail*) im Internet. Dazu definiert es wie Mail-Systeme interagieren und welches Format Steuermeldungen haben müssen. Die genaue Definition findet man in RFC 821 (Request For Comments). Wie HTTP benutzt auch SMTP den ASCII-Zeichensatz zur Kommunikation. Zur Zustellung der E-Mails baut der Quellrechner eine TCP-Verbindung zum Port 25 des Zielrechners auf. Dieses Protokoll wird heutzutage von mehreren Gemeinden eingesetzt, um Anträge oder Anfragen von Bürgern an die Stadtverwaltung zu übermitteln. Da der gesamte Inhalt einer E-Mail aber unverschlüsselt über die Server im Internet geleitet wird, können leicht vertrauliche Daten abgefangen oder verfälscht werden. Abhilfe bietet hier eine Verschlüsselung der Daten, was aber vorraussetzt, dass die Stadtverwaltung und der Antragsteller über zertifizierte Schlüssel verfügen müssen (vgl. Abschnitt 3. 5. 3).

3. 2 Datenbank und Schnittstellen

3. 2. 1 Begriffe

Als Datenbank wird das komplette *Datenbank-Management-System* (*DBMS*) bezeichnet. Genaugenommen bezeichnet der Begriff Datenbank nur die Sammlung der gespeicherten Daten, die von Anwendungen benötigt werden. Bei einem DBMS handelt es sich um ein standardisiertes Software-System zur Definition, Verwaltung, Verarbeitung und Auswertung der Datenbank-Daten. Die wichtigsten Operationen sind Einfügen, Löschen, Ändern von Datensätzen sowie eine Auswahloperation zur Abfrage von Daten.
Man unterscheidet heutzutage drei verschiedene Datenbankmodelle: relational, objekt-relational und objekt-orientiert. Die grundlegende Datenstruktur des relationalen Modells ist die Tabelle, welche eine Menge von Tupeln mit gleichen Attributen und unterschiedlichen Attributwerten enthält. Die Eindeutigkeit der Tupel wird durch eine Auswahl ausgezeichneter Attribute (den Primärschlüssel) sichergestellt. Das objekt-orientierte Datenmodell enthält als

Grundlage Objekte mit Attributen und Methoden, deren Instanzen zur Speicherung der Daten dienen. Das objekt-relationale Modell verbindet die effizienten Datenstrukturen des relationalen Modells mit der Möglichkeit eigene Objekte als Attribute zu definieren.
Die größte Verbreitung haben Datenbanken, die auf dem relationalen Datenmodell basieren. Objekt-orientierte Systeme haben sich bisher kaum durchsetzen können. Um auf deren Vorteile nicht verzichten zu müssen, erweitern viele Hersteller relationaler Datenbanksysteme ihre Produkte zu objekt-relationalen Datenbanksystemen (beispielsweise IBM DB2, Oracle 8i).
Als von *American National Standards Institute* (*ANSI*) und ISO standardisierte Anfragesprache für relationale Datenbanken hat sich die *Structured Query Language* (*SQL*) durchgesetzt (vgl. [DD98]). Der erste Standard stammt aus dem Jahre 1987, 1992 wurde der SQL-2 bzw. SQL-92 Standard verabschiedet, welcher heute überwiegend Verwendung findet. In ihm sind die grundlegenden Operationen festgeschrieben, die für die Arbeit mit einem relationalen System notwendig sind. Zu jeder Funktion gibt es bestimmte Schlüsselwörter, von denen die wichtigsten in folgender Tabelle 3-1 kurz beschrieben werden.

Schlüsselwort	Funktion
SELECT	Auswahl von Datensätzen, die bestimmten Kriterien entsprechen
INSERT	Einfügen von neuen Datensätzen
UPDATE	Änderung von bestehenden Datensätzen
DELETE	Löschen von Datensätzen
CREATE	Erstellung von Datenbankobjekten wie Tabellen oder Nutzern
DROP	Löschen von Datenbankobjekten

Tabelle 3-1: SQL-Funktionen

3. 2. 2 Schnittstellen

Je nachdem welcher Typ von Applikation zum Einsatz kommt, werden auch die verwendeten Datenbankschnittstellen variieren. Bei reinen Windows-Anwendungen drängt sich natürlich die Verwendung von ODBC auf. Wird die Anwendung in Java entwickelt so wird JDBC zum Einsatz kommen. Für den Fall, dass die Funktionalität durch eine Skriptsprache realisiert wird, wer-

den die von der Skriptsprache bereitgestellten Funktionen genutzt.

ODBC

Open Database Connectivity (*ODBC*) wurde 1992 von Microsoft entwickelt. Es handelt sich hierbei um eine Software-Schnittstelle, die den Zugriff aus einem Anwendungsprogramm auf verschiedene Datenbanken gewährleistet. Aufgrund der Entwicklungsfirma kommt ODBC vorwiegend in Windows-Umgebungen zum Einsatz.

JDBC

Die *Java DBC* (*JDBC*) stellt das Gegenstück zu ODBC dar. Es ist eine Schnittstelle, mit deren Hilfe man aus Java-Anwendungen heraus auf die verschiedenen Datenbanken zugreifen kann. Die Bearbeitungs- und Benachrichtigungskomponente des zu entwickelnden Systems kann auch als Java-Applikation realisiert werden, ohne das man die Prämisse "webbasiert" verwerfen muss.

ASP

Unter Verwendung von *Active Server Pages* (*ASP*), welche von Microsoft entwickelt wurden, können HTML Seiten dynamisch gestaltet werden. Es handelt sich hierbei um eine Skriptsprache, bei der die Skripte, die die HTML-Ausgabe erzeugen, vom Webserver interpretiert werden. Als Grundlage für die Ausgabe können zum Beispiel auch Datenbankinhalte dienen. Voraussetzung für die Verwendung von ASP ist der Einsatz des Microsoft Internet Information Servers.

PHP

Bei *PHP: Hypertext Preprocessor* (*PHP*) handelt es sich um eine Skriptsprache, die server-seitig in HTML eingebettet ist. Den Anfang der Entwicklung von PHP machte 1994 der Däne Rasmus Lerdorf. Dieser wandte sich von den in Perl geschriebenen CGI-Programmen ab und implementierte diese auf Grund der besseren Performance zunächst in C. Die Syntax ist dementsprechend an C angelehnt. Eine Beschreibung von PHP erfolgt in Abschnitt 4. 1. 2.

3. 3 Webserver

Damit ASP oder PHP zum Einsatz kommen können, muss ein Webserver zur Verfügung stehen. Dieser besteht aus folgenden Komponenten (vgl. Abb. 3-1):

- einem *Listener*, welcher auf Client-Anfragen wartet und diese weiterleitet,
- einem *Broker*, der die Anfragen bearbeitet und letztendlich dem Client antwortet (meist eine HTML-Datei ausgibt),
- und der im zu entwickelnden System wichtigen *Schnittstelle* mit eigenem Interpreter, welche durch Skripte gesteuert Verbindungen zur Datenbank aufbaut und Inhalte aus der Datenbank an den Broker zur Ausgabe weiterleitet.

Die im letzten Punkt genannte Schnittstelle ermöglicht die Verwendung dynamisch generierter HTML-Seiten. Neben dem Apache Webserver [APA01] sind auch der Microsoft Internet Information Server (MS IIS) und andere dazu in der Lage. Dabei werden unterschiedliche Konzepte verfolgt. Werden beim MS IIS meist sogenannte *Active Server Pages* (*ASP*) interpretiert, kann man beim Apache Webserver auch andere Erweiterungen wie Perl oder PHP einsetzen. Diese Erweiterungen sind für die Datenbankzugriffe und das Erzeugen des HTML-Quellcodes zuständig. Am häufigsten kommt im Internet der Webserver von Apache zum Einsatz (vgl. [NC01]), welcher auch als Kern für kommerzielle Produkte wie dem *Oracle Internet Application Server* (*IAS*) dient. Als Grund dafür ist einmal zu sehen, dass es sich um einen recht sicheren und ausgereiften Webserver handelt, der dem Open Source Modell zuzuordnen ist. Er ist für die unterschiedlichen Betriebssysteme wie Linux, Solaris, Microsoft Windows Familie und andere frei erhältlich. Durch den offengelegten Quellcode können Sicherheitslöcher schneller gefunden und ausgebessert werden als bei kommerziellen Produkten. Auch bezüglich der Zuverlässigkeit unter hoher Last hat er gegenüber anderen Mitbewerbern Vorteile. Dies hängt aber auch vom verwendeten Server-Betriebssystem ab. Eine sinnvolle Kombination wäre Linux/Apache oder Solaris/Apache. Einen Geschwindigkeitsvorteil erzielt man mit der Einbindung des PHP-Moduls in den Webserver (vgl. 3. 2. 2).

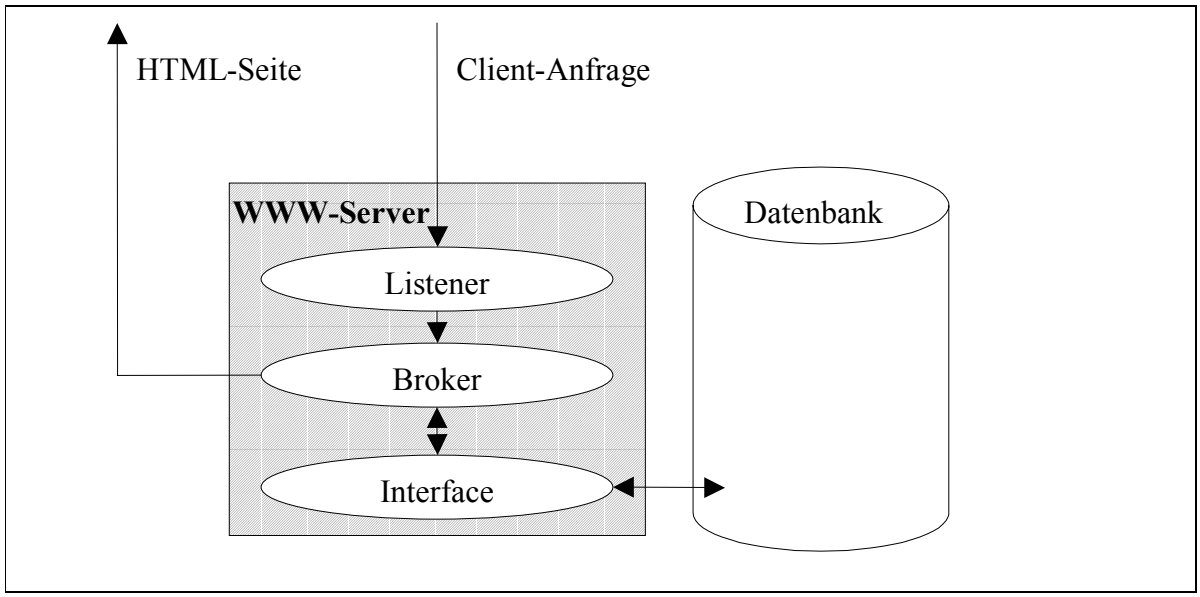

Abb. 3-1: Datenfluss durch die WWW-Server-Software

3. 4 Gestaltung

In Anbetracht der Tatsache, dass in der vorliegenden Arbeit ein web-basiertes System geschaffen werden soll, welches im Browser läuft, kommen als Oberfläche eigentlich nur zwei Varianten in Frage: Java-Applet oder HTML. Während bei der Realisierung mittels Java ansprechendere und umfangreichere Oberflächen möglich sind, hat das Paket aus HTML und einer Skriptsprache vor allem bei der Geschwindigkeit große Vorteile. Das entscheidende Argument gegen ein Java- oder jedes andere fest programmierte System ist aber, dass in diesen Fällen für jeden Antrag eine eigene Oberfläche für die Bearbeitung der Antragsdaten zusätzlich zu den HTML-Formularen für die Bürger programmiert werden muss.
Ein Großteil der Funktionalität läuft unsichtbar ab. Eine Schnittstelle zum Mitarbeiter wird nur bei der Recherche, der Administration und der Präsentation der Antragsdaten benötigt.

3. 5 Sicherheit

3. 5. 1 Unverschlüsselte Übertragung mittels Browser

Schon zu Beginn der Übertragung der Daten vom Rechner des Bürgers zum Server in die Verwaltung sind diese gefährdet. Hier ist man aber an die Nutzung der vorhandenen Kommunikationskomponenten (das WWW mit WWW-Servern, Browsern und den zugehörigen Protokollen) aus Akzeptanzgründen gebunden. Hingegen könnte man zum Beispiel innerhalb der Verwaltung vorschreiben, dass ein Java-Client zur Bearbeitung der Antragsdaten benutzt werden muss. Aber innerhalb des Verwaltungsnetzes sind die Daten nicht so großen Gefahren ausgesetzt wie im (anonymen) Internet. Hier kann man genau abgrenzen (z.B. durch einen Begrenzungs-Router), welche Datenpakete das Verwaltungsnetz verlassen dürfen. Deshalb kommt für die Vorgangssteuerung sehr wohl eine Browser-Lösung in Frage.
Es bestehen grundsätzlich zwei Möglichkeiten der Übergabe der Daten an ein Skript. Die erste Variante wäre, die Daten mit der sogenannte GET-Methode des HTTP via URL zu übergeben:

http://www.domain.de/skript.php?datenfeld1=Daniel&datenfeld2=Heinze

Die zu übergebenden Variablennamen und deren Werte werden - getrennt von einem „?“ - an den normalen URL eines Skripts angehängt. Variablenname und Wert der Variable werden durch ein „=“ getrennt. Werden mehrere Variablen übergeben, so werden die Variablen untereinander durch „&“ separiert. Die Aneinanderreihung sieht dann folgendermaßen aus:

?variablenname1=wert1&variablenname2=wert2& ... &variablennameN=wertN

Die Daten werden vom WWW-Server in der Umgebungsvariablen QUERY_STRING gespeichert.
Viele Server protokollieren den URL eines Request. Bei Verwendung der GET-Methode werden damit dann auch die Daten abgespeichert. Abgesehen davon dass diese Form der Datenübergabe für persönliche Daten nicht geeignet erscheint, weil alle Angaben im URL zu lesen sind (mitunter auch für andere), gibt es auch eine Beschränkung für die Länge des URL. Nach eigenen Tests mit einem Apache Webserver und einem Netscape Communicator 4.7 beträgt diese 8202 Zeichen. Die effektiv nutzbare Anzahl von Zeichen ergibt sich dann aus:

N_{eff} = 8202 - [Anzahl Zeichen Rechneradresse + 1] - [Anzahl Zeichen Skriptname] - [Anzahl Zeichen Variablennamen + 2 * Anzahl Variablen]

Für die meisten Anwendungen sollten die restlichen meist über 7000 Zeichen ausreichen. Man muss jedoch beachten, dass es auch Formulare gibt, in denen Mitteilungen in Texteingabefeldern gemacht werden können. So könnten zum Beispiel ein Anschreiben oder andere längere Texte schnell N_{eff} überschreiten.

Bei der zweiten Form der Datenübermittlung, der HTTP-Methode POST, werden die Daten nicht im Logfile des Webservers gespeichert und sind daher auch nicht für Benutzer (Administratoren und Personen mit eigenem Login) des WWW-Servers einsehbar. Der WWW-Server liest die Daten dabei von der Standardeingabe. Diese Methode ist für unterschiedliche Daten mit variabler Länge geeignet. So können zum Beispiel auch Bilder (Skizzen bei Bauanträgen) übermittelt werden.

Dennoch bietet auch diese Variante keinen Schutz gegen das „Abhören" der Pakete mit sogenannten „*Paketsniffern*". Die Inhalte der Formulare werden unverschlüsselt zum Webserver transportiert und können von jedem Rechner, der sich auf dem Weg zwischen Client und WWW-Server befindet, ausgelesen und protokolliert werden.

3. 5. 2 Secure Socket Layer (SSL)

Die verbreitetste Art, Daten sicher über das Internet zu verschicken, ist die Nutzung des von Netscape entwickelten Sicherheitsprotokolls *Secure Socket Layer* (*SSL*) ([NSL01]). Dies ist ein hybrides Verfahren aus symmetrischer und asymmetrischer Verschlüsselung. SSL ist zwischen der Transportschicht (TCP) und der Anwendungsschicht (http, ftp, Telnet) einzuordnen. Für die Anwendungen selbst ist SSL unsichtbar, was bedeutet, dass keine Anpassungen erforderlich sind. Mit Hilfe von SSL wird eine Vertraulichkeit erreicht, die durch die verschlüsselte Übertragung der Daten zwischen WWW-Server und Client entsteht.

Der Server benutzt dabei einen Schlüssel (Key), den er bei einer offiziellen Zertifizierungsstelle (CA) registriert hat, damit die CA die Korrektheit dieses Schlüssels bestätigen kann. Somit kann jeder überprüfen, ob der Rechner (WWW-Server), von dem er den Schlüssel für eine Session

bekommen hat, wirklich der ist, für den er sich ausgibt, und er kann auch sicher sein, dass die Verbindung gegen Abhören geschützt ist.
Weiterhin ist es von Vorteil, dass der Bürger keine zusätzlichen Programme benötigt, die er auf seinem Rechner installieren muss, sondern er kann ganz normal über seinen Browser die Eingaben tätigen, da die heutzutage üblichen Browser SSL-Verbindungen unterstützen. Statt eine Verbindung nach dem Hypertext Transfer Protokoll zum WWW-Server aufzubauen, wird diese nach dem HTTPS geknüpft.
Damit SSL eingesetzt werden kann, muss der WWW-Server dafür konfiguriert werden. Bei der Verwendung des Apache Webservers kann auf die von Eric Young und Tim Hudson entwickelte SSL-Bibliothek namens SSLeay [HY98] zurückgegriffen werden.

3. 5. 3 Pretty Good Privacy (PGP)

Eine weitere Möglichkeit sensitive Daten (wie Antragsdaten) sicher über das Internet oder andere offene Netze zu verschicken ist der Einsatz von *Pretty Good Privacy* (*PGP*, [PGP01]), zu deutsch: ziemlich gute Privatsphäre. Dieses von Phil Zimmermann 1990 entwickelte Verfahren zur sicheren Übertragung von Daten über ungesicherte Netze bietet neben der Wahrung der Datenintegrität auch die Möglichkeit der Signatur. So wird sichergestellt, dass der Inhalt der Nachricht den Empfänger unverfälscht erreicht und der Empfänger genau feststellen kann, von wem die Nachricht stammt. Dies wird durch die Verwendung eines Schlüsselpaares erreicht. Jeder Teilnehmer an diesem Verfahren besitzt zwei Schlüssel: einen privaten und einen öffentlichen. Der private Schlüssel bleibt beim Besitzer während der öffentliche Schlüssel den anderen Nutzern zugänglich gemacht wird. Dies kann über einen Key-Server (ein Server im Internet, der öffentliche Schlüssel von verschiedenen Personen oder Institutionen zum download anbietet) oder einfach durch das Verschicken in einer E-Mail geschehen. Ein geheimer Schlüsselaustausch, wie er bei symmetrischer Verschlüsselung notwendig ist, erübrigt sich. Dies folgt aus der Forderung, die alle Public-Key Verfahren erfüllen müssen, nämlich dass es unmöglich sein soll, den privaten aus dem öffentlichen Schlüssel zu berechnen.
Dieses Verfahren hat gegenüber SSL den Nachteil, dass jeder Benutzer ein Schlüsselpaar benötigt. Im Falle einer Gemeinde bedeutet dies, dass jeder Bürger, der einmal einen Antrag über das Internet an seine Stadtverwaltung stellen möchte, ein Schlüsselpaar erzeugen, sich den öffentlichen Schlüssel der Stadtverwaltung besorgen und im Gegenzug der Stadtverwaltung seinen

öffentlichen Schlüssel zugänglich machen muss. Auch der Erwerb oder das Herunterladen der Software zum Erstellen und Benutzen der Schlüssel schreckt viele Bürger ab. Die Erfahrung zeigt, dass die Mehrzahl der Internet-Benutzer Dienste nicht in Anspruch nehmen, bei denen sie zusätzliche Programme kaufen bzw. herunterladen, installieren und benutzen müssen. In diesem Bereich fehlt einfach noch ein Standard, der von möglichst vielen Anbietern unterstützt wird.

Neben der Softwarelösung wie zum Beispiel PGP gibt es auch andere Ansätze, bei denen der private Schlüssel auf einer Chipkarte gespeichert wird, die bei Bedarf in ein Lesegerät geschoben werden muss, damit die Nachricht verschlüsselt werden kann. Diese Lösung ist zwingend notwendig, wenn der Rechner, an dem ein Benutzer verschlüsselte oder signierte Nachrichten verschickt, von mehreren Personen genutzt wird. Eine andere Möglichkeit in diesem Zusammenhang ist die Speicherung des privaten Schlüssels auf einer Diskette oder einem anderem Wechselmedium statt auf der Festplatte. Dies ist aber unhandlicher.

3. 5. 4 Gegenwärtiger Stand

Aufgrund des Nichtvorhandenseins von Mitteln (Ausrüstung) muss man heutzutage noch Einschränkungen bei der Antragstellung über das Internet in Kauf nehmen. Bei vielen Anträgen ist es zwingend notwendig, dass sich der Absender (Antragsteller) einwandfrei identifiziert, so wie er es herkömmlicherweise per Unterschrift macht. Die Gemeinden gehen da unterschiedliche Wege. Die einen bieten erst gar nicht solche Dienste an, bei denen eine eindeutige Identifizierung (persönliches Vorsprechen im Amt) zwingend notwendig ist - wie zum Beispiel die Stadt Beelitz, die sich auf solche Dienste wie "Hundesteuer: An- und Abmeldung" oder "Einzugsermächtigung für kommunale Steuern" (vgl. [Bee00]) beschränkt, welche im Browser ausgefüllt und dann ausgedruckt werden können, bevor sie mit der Post an die Stadtverwaltung gesendet werden müssen.

Die Stadt Meißen wiederum geht einen ganz anderen Weg. Die potenziellen Nutzer ihres Systems müssen sich registrieren. Dies geschieht folgendermaßen: Der Bürger füllt im Internet ein Formular mit seinen Daten aus (vgl. [MEI01]). Schickt er dieses Formular ab, so wird die zuständige Stelle im Amt benachrichtigt, welche nun überprüfen muss, ob ein Bürger mit dem genannten Namen an dem angegebenen Ort wohnhaft ist. Ist dies der Fall, so kann der Angestellte einen vorgefertigten Brief ausdrucken, der dann per normaler Post dem Antragsteller zugestellt

wird. Erst mit diesem Brief erhält man das Passwort, mit dessen Hilfe man im Internetauftritt der Stadt Meißen online Anträge stellen kann. Man hofft, dass diese umständliche Prozedur umgestellt werden kann, wenn in Deutschland die digitale Signatur verbreitet angewandt wird.

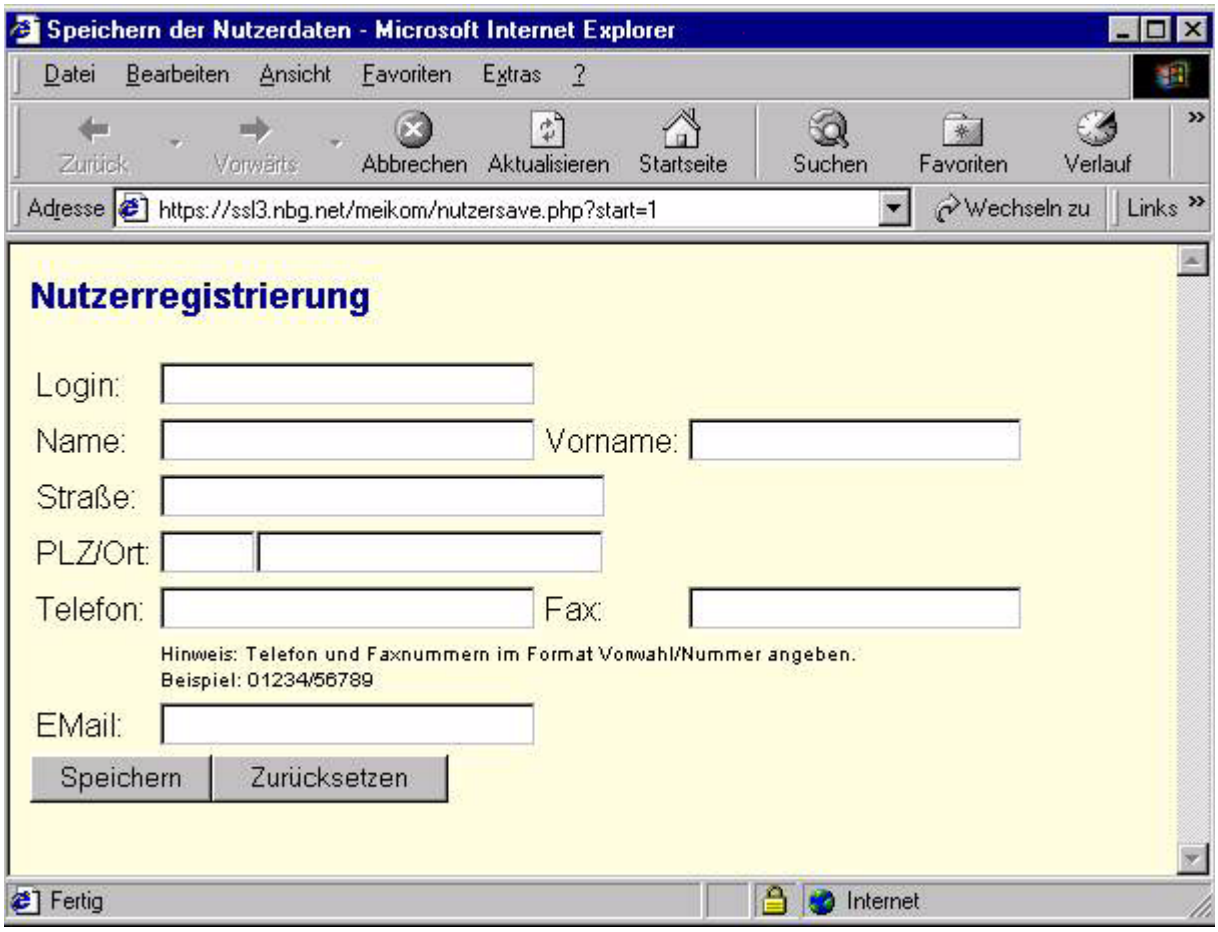

Abb. 3-2: Registrierung Meißen aus [Mei01]

3. 5. 5 Die digitale Signatur

Möchte man über das Internet rechtsverbindliche Verwaltungsakte durchführen, so bedarf es eines Ersatzes der handschriftlichen Unterschrift aus dem normalen Leben.

Es gibt zwei Punkte, die man beim Datenaustausch beachten sollte, sei es über koventionelle Übertragungswege wie Post oder elektronische Übertragungsmedien.

Zum einen ist das die Gewissheit, dass die Daten nur von einer bestimmten Person geschrieben und abgesendet wurden (Authentizität). Zum anderen muss gewährleistet sein, dass die Daten auf dem Weg vom Absender zum Empfänger nicht verändert werden können, ohne dass der

Empfänger dies merkt (Integrität). Bei üblichen elektronischen Übermittlungen von Dokumenten wie E-mail, Fax und Archivierung auf elektronischen Datenträgern können die Daten durch gezielte Manipulation verändert werden, ohne dass dies unmittelbar feststellbar ist.
Soweit es sich um originale Dokumente (keine Kopien) in Papierform handelt, reicht die eigenhändige Unterschrift unter dem Schriftstück, um sicher zu sein, wer der Absender ist. Die eigenhändige Unterschrift gilt als rechtsverbindlich. Aber auch diese Unterschrift ist manchmal unbefriedigend. Zum Beispiel kennt jemand, der noch nie die Handschrift eines Vertragspartners gesehen hat, auch nicht dessen Unterschrift. Es kann somit auch eine völlig andere Person unterschrieben haben.
Hier ist man mit der digitalen Signatur sogar schon weiter. Es ist jederzeit möglich eine digitale Signatur innerhalb kürzester Zeit, das heißt wenige Sekunden, online überprüfen zu lassen. Infolgedessen hat man schnell die Gewissheit, dass auch die richtige Person unterschrieben hat. Deshalb bietet ein elektronisches Dokument, welches mit Hilfe einer geeigneten Methode elektronisch unterschrieben worden ist, eine höhere Fälschungssicherheit als Dokumente in Papierform. Dabei gilt als geeignete Methode vor allem die gesetzlich anerkannte digitale Signatur welche der eigenhändigen Unterschrift durch einen Zusatz zum Bürgerlichen Gesetzbuch (BGB) gleichgestellt worden ist. Bis letztendlich jeder Bürger sein persönliches Siegel in Form einer dieser Chipkarten besitzt werden nach den Worten des Referatsleiter Digitale Signatur bei der Regulierungsbehörde für Telekommunikation und Post, Jürgen Schwemmer, noch fünf bis sechs Jahre vergehen. Also spätestens 2006 werden keine Rechner ohne ein entsprechendes Lesegerät für diese Chipkarten verkauft werden.
Heutzutage gibt es vereinzelt schon Anwender der Chipkarten. Momentan werden diese meist zur Abwicklung von Bankgeschäften über das Internet benutzt. Die Kosten für die Bereitstellung der Lesegeräte und der Chipkarten werden noch immer auf die Kunden übertragen. Das ist ein Grund, weshalb die Verbreitung nur schleppend vorangeht. Weitaus wichtiger ist jedoch, dass es bisher noch keine konkreten Antworten auf wichtige Fragen im Zusammenhang mit der digitalen Signatur gibt. Die digital signierten Dokumente müssen über einen längeren Zeitraum aufbewahrt werden. Konkrete Vorgaben auf welche Art und Weise dies umzusetzen ist, gibt es bis heute nicht. Weiterhin wird derzeit diskutiert, wie man digital signierte Dokumente über mehrere Jahre hinaus schützt, wenn die anfangs angewandten kryptographischen Algorithmen längst veraltet sind. Nach dem Signaturgesetz sollen alle Dokumente zyklisch nach dem neuesten Stand der Sicherheitstechnik übersigniert werden (vgl. [CZ4101], S. 4). Das heißt jedes

Unternehmen und jede Person, welche digital signierte Dokumente verwaltet, muss nach noch nicht festgelegten Zeitintervallen alle Dokumente mit einem neueren kryptographischen Algorithmus zusätzlich signieren. Welchen Aufwand dies nicht nur für die Unternehmen sondern auch für die normalen Bürger bedeutet, die zum Beispiel auch ihre digital signierten Kaufverträge auf den neuesten Sicherheitsstand bringen müssen, ist noch nicht endgültig abzuschätzen, jedoch werden viele Bürger dies "verschlafen". Deshalb ist eine zentrale Archivierung auch privater Dokumente im Gespräch. Damit würden diese Arbeiten von den verwaltenden Unternehmen durchgeführt werden (vgl. [CZ4101], S. 10).

3. 6 Datenspeicherung

Sowohl alle Antragsdaten als auch alle Informationen zur Steuerung der Daten innerhalb der Verwaltung werden auf einem Server in einer Datenbank oder in Dateien gespeichert. Damit ist die Sicherheit dieses Rechners von höchster Bedeutung.
Es ist darauf zu achten, dass keine Person von außerhalb sich Zugriff auf die Daten verschaffen kann. Dies ist nicht allein damit getan, dass man nur bestimmten Personen ein Login und Passwort gibt, diese kann der Angreifer nämlich von außerhalb mit viel Fleiß herausbekommen. Vielmehr sollte genau protokolliert werden, wann wer von wo versucht hat auf den Server zuzugreifen. Missglückte Anmeldeversuche sollten automatisch dem zuständigen Administrator gemeldet werden.
Es wäre zu empfehlen ein angrenzendes internes Netz mit Hilfe einer Firewall nach außen abzusichern. Hier wäre der Einsatz eines Dual Home Bastion Host angebracht. Dabei befindet sich der WWW-Server in einer „*Demilitarisierten Zone*“ und das interne Netz wird von einem Bastion Host abgesichert (vgl. Abschnitt 3. 7. 3).

3. 7 Zugriffssteuerung

3. 7. 1 Datenbankzugriffe

Damit nicht jeder beliebige Mitarbeiter der Verwaltung Zugriff auf alle Daten in der Datenbank

hat, werden Rollen vergeben. Dadurch ist es möglich den Bearbeitern nur einen Ausschnitt der Datenbank zu präsentieren, nämlich den Teil, für den sie wirklich zuständig sind. Die anderen Daten bleiben für sie unerreichbar, da sie nicht die dafür nötigen Zugriffsrechte besitzen. Die Mitarbeiter selbst bekommen nicht einmal ein Login auf der Datenbank. Für die Eingabe und die Ausgabe der Daten in die beziehungsweise aus der Datenbank sind die PHP-Skripte zuständig. Diese müssen sich gegenüber der Datenbank authentifizieren. Das wird dadurch erreicht, dass einer Authentifikationsprozedur der Loginname und das Passwort, welche zum Bearbeiten der Daten in der Datenbank notwendig sind, übergeben werden und diese Authentifikationsprozedur einen Handler zurückgibt. Welcher Loginname und welches Passwort dabei übergeben werden hängt davon ab, wer sich gerade im System angemeldet hat. Anhand dessen wird entschieden, welche Rechte der Nutzer auf der Datenbank hat. Diese Informationen sind natürlich auch in der Datenbank hinterlegt. Den Zugriff darauf hat das Skript über einen anderen temporären Handler.

3. 7. 2 Zugangskontrolle für Antragsbearbeitung

Der Schutz vor unbefugtem Zugriff auf Antragsdaten wird dadurch erreicht, dass jeder Mitarbeiter, der Anträge bearbeiten muss, ein Login/Passwort - Paar bekommt. Diese muss er eingeben, bevor er überhaupt etwas bearbeiten kann. Wird das System mit Java realisiert, so erfolgt die Eingabe des Logins und des Passworts direkt am Client. Dieser vergleicht die Eingabe mit den Daten vom Server. Stimmen sie überein, so wird dem Mitarbeiter Zugriff gewährt.
Bei einer Bearbeitung über einen Browser läuft dies etwas anders. Auch hier muss der Mitarbeiter sein Login und sein Passwort eingeben. Die Skripte verifizieren die Eingabe mit dem in der Datenbank gespeicherten Login/Passwort und nach erfolgreicher Überprüfung kann der Mitarbeiter die Angelegenheiten erledigen, für die er im System zuständig ist. Dies kann zum Beispiel die Administration der Anträge sein, die Bearbeitung der Antragsdaten oder die Recherche darauf. Anders als bei einer Java-Client-Lösung bleibt die Verbindung zur Datenbank nicht bestehen. Nach jedem neuen Skriptaufruf oder jedem Klick auf einen Link müsste der Mitarbeiter sich neu anmelden, da HTML ein zustandsloses Protokoll ist. Abhilfe schafft hier der Einsatz von Hilfsmitteln wie Cookies oder Sessions, die zum Funktionsumfang von einigen Skriptsprachen gehören. Bei Sessions werden die Daten auf dem Server als Cookie hinterlegt. Die Zuordnung erfolgt dann durch eine Session ID, die im URL übergeben werden kann (vgl.

[PHP01]).

Eine wichtige Rolle für die Sicherheit spielt auch die Ablage der Skripte. Werden zum Beispiel alle Skripte in Verzeichnissen abgelegt, auf die vom WWW-Server zugegriffen werden kann, so sollten dort keine Login/Passwort-Paare enthalten sein. Damit die Zugriffsregelung funktioniert, müssen einige Skripte eine Verbindung zur Datenbank mit einem internen Login/Passwort-Paar aufbauen. Damit nicht jeder dieses interne Login/Passwort-Paar aus den Skripten herauslesen kann, müssen diese Teile des Skriptes in ein höheres Verzeichnis abgelegt und in die Skripte eingefügt werden (z.B. durch INCLUDE).

3. 7. 3 Firewall

Für die Absicherung der Server vor neugierigen oder böswilligen Internet-Nutzern sollte ein Firewall-System installiert werden. Je nachdem für welche Methode man sich entscheidet kommen auch hier noch einige Hardwarekomponenten (wie Router oder Firewall-Rechner) zusammen. Im einfachsten Fall installiert man die Firewall-Software auf einem der schon vorhandenen Server, was natürlich nicht gerade den besten Schutz bietet. Normalerweise wird man ein Firewall-System mit Dual Home Bastion Host oder ein System mit abgesichertem Zwischennetz einsetzen. In jedem Fall muss gewährleistet sein, dass die Skripte, welche von den Mitarbeitern in der Stadtverwaltung zum Bearbeiten der Anträge benutzt werden, auf keinen Fall von außerhalb ausgeführt werden können.

Damit ergeben sich dann folgende Schemata:

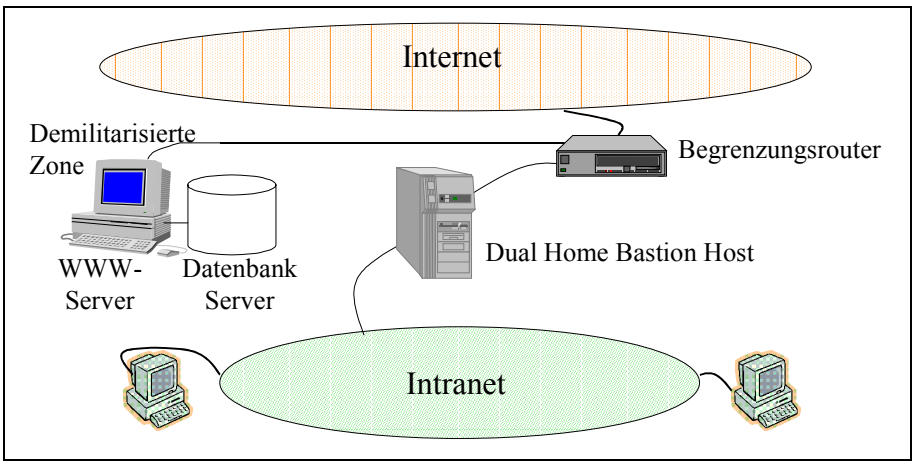

Abb. 3-3: Dual Home Bastion Host

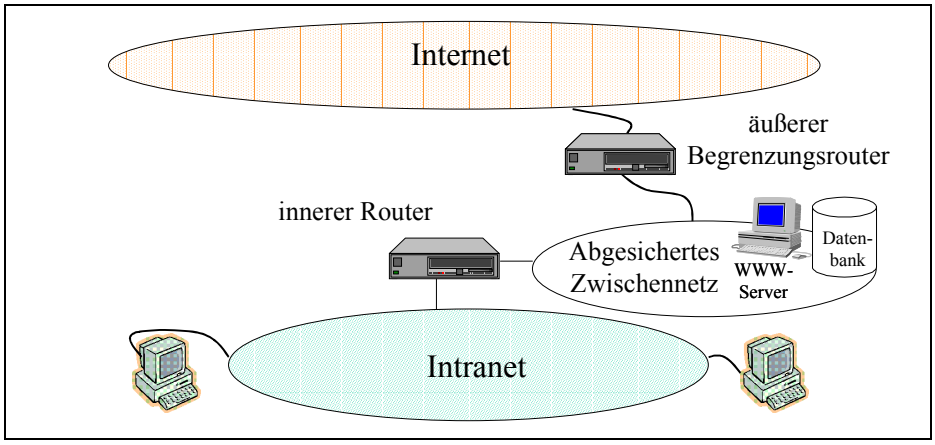

Abb. 3-4: Abgesichertes Zwischennetz

4 Architektur/Implementierung

4. 1 Gewählte Produkte

4. 1. 1 Überblick

Da das System eine Erweiterung der Arbeiten von Steinkopf [Ste99] und Wiebigke [Wie99] darstellt, wird bis auf einige Änderungen das dort erarbeitete Grundgerüst von Hard- und Software-Komponenten übernommen.
In [Ste99] wird ein Mehrprozessorsystem mit UltraSparc II Prozessoren der Firma Sun zugrundegelegt. Dieses bietet eine einfache Erweiterungsmöglichkeit bei höherem Ressourcenbedarf durch die Integration von weiteren Prozessoren. Der Hauptspeicher wurde mit 512 MB angegeben. Das von Steinkopf beschriebene System beherbergte den WWW-Server und das Datenbank-Management-System.
Neben der Hardware wurden auch grundsätzliche Aussagen zur zu verwendenden Software getroffen. So dient Solaris (aktuelle Version 8) von Sun als Betriebssystem. Als Entwicklungsserver dient jedoch ein AMD Athlon Thunderbird 1,2 GHz Rechner (256MB RAM) mit Linux und Microsoft Windows Me Betriebssystem, womit das gesamte System bei maximal fünf gleichzeitigen Nutzern ausreichende Performance bietet, so dass die von Steinkopf genannte Rechnerkonfiguration zwar als Richtgröße für praktische Einsätze anzusehen ist, jedoch stark von dem zu erwartenden Zugriffsaufkommen und dem Budget abhängt.
Der schon von Wiebigke und Steinkopf eingesetzte Webserver Apache konnte seine Beliebtheit bis heute noch ausbauen [NC01], er gilt dank der Open Source Strategie seiner Entwickler als sehr sicher und stabil. Abbildung 4-1 zeigt eindrucksvoll die Marktführerschaft dieses Produkts.

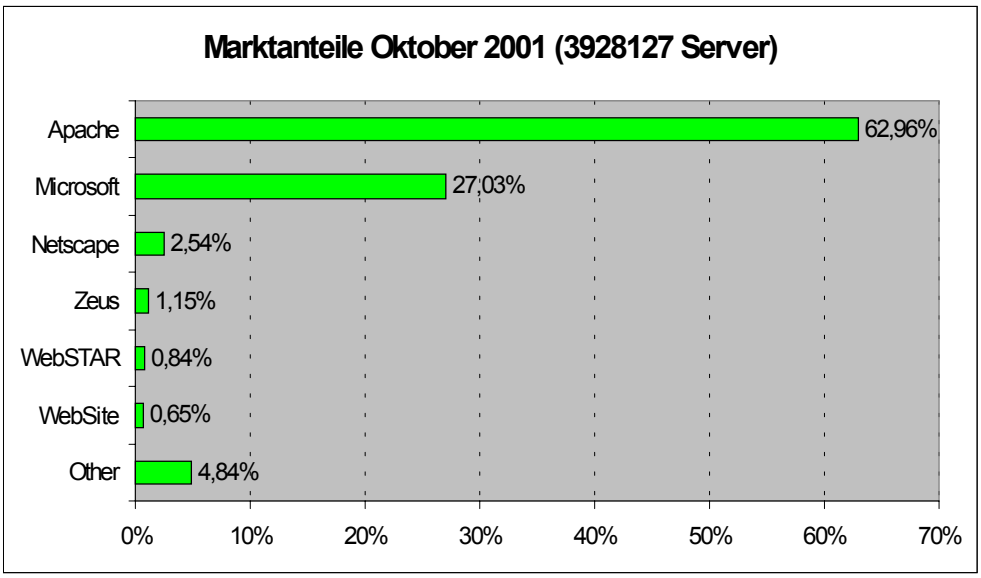

Abb. 4-1: Webserver - Marktanteile Oktober 2001 nach [SeSp01]

Auch in der vorliegenden Arbeit wird der Apache in der Version 1.3.22 benutzt. Als Interface zur Datenbank kommt wie gehabt PHP, nun in der Version 4.0.6 zum Einsatz. Als Datenbank-Management-System wird MySQL 3.23 eingesetzt.
Zur Betrachtung der (dynamisch generierten) HTML-Seiten ist ein Browser einzusetzen, an den aber - wie an den E-Mail-Client - keine besonderen Anforderungen gestellt werden. Somit kann der Nutzer mit seinem favorisierten und vertrauten Produkt arbeiten. Zur Benachrichtigung der Mitarbeiter über den Eingang von neuen Antragsdaten wird ein Mailserver benötigt. Für die vorliegende Arbeit wird der in der SuSE Linux Distribution [Sus01] mitgelieferte sendmail benutzt.

4. 1. 2 Beschreibung PHP

Um sich die Arbeit zu erleichtern, entwickelte der Däne Rasmus Lerdorf eine Authoring-Utility-Sammlung. Diese bestand aus einem Parser mit einigen Makros und Werkzeugen. Ein von ihm früher entwickeltes Tool, ein Formularinterpreter mit mSQL-Unterstützung namens FI verband er 1995 mit diesem neuen Paket zu PHP/FI 2, welches aus einem yacc Parser und einem selbstentwickelten lexikalischen Analysator bestand.
PHP wurde schnell populär und es fanden sich einige Leute, die zusammen mit Lerdorf ein gut

organisiertes Team bildeten, welches die Entwicklung effektiver werden ließ. Die Basis für PHP 3 bildet der von Zeev Suraski und Andi Gutmans umgeschriebene Parser. Seit diesem gab es keine gravierenden Änderungen. Dieses Produkt wurde seit der ersten Version nach dem Open Source Modell entwickelt. Aktuelle Version ist PHP 4.0.5. Einen Grund für die Popularität von PHP ist die Unterstützung von vielen Datenbanken, auf die mit einfachen Techniken zugegriffen werden kann. Folgende Auflistung soll den Umfang kurz umreißen:
Oracle, Adabas D, Sybase, FilePro, mSQL, MySQL, Informix, Solid, dbase, ODBC, PostgreSQL, Unix dbm, Velocis.

PHP kann sowohl als CGI ausgeführt werden als auch im Apache Webserver als Modul integriert werden. Letzteres resultiert in einem erheblichen Geschwindigkeitsvorteil, da die ansonsten obligatorischen fork() und exec() Sequenzen der CGI-Variante wegfallen. Die Dateien, welche den PHP-Code enthalten, werden anhand ihrer Dateinamenendung (zum Beispiel .php) erkannt und beim Aufruf über den Browser nicht wie normale HTML-Dateien ausgegeben sondern nach dem PHP-Code (welcher sich in den PHP-Tags befindet) durchsucht. Dieser wird ausgeführt und das Ergebnis in HTML-Form ausgegeben. Ein kleines Beispiel soll die Funktionsweise verdeutlichen:
Der Quelltext der Datei helloworld.php:

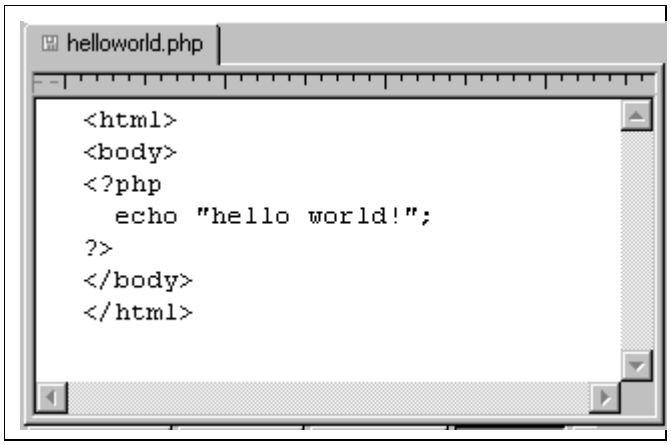

Abb. 4-2: helloworld.php Skript

Die Ausgabe im Browser:

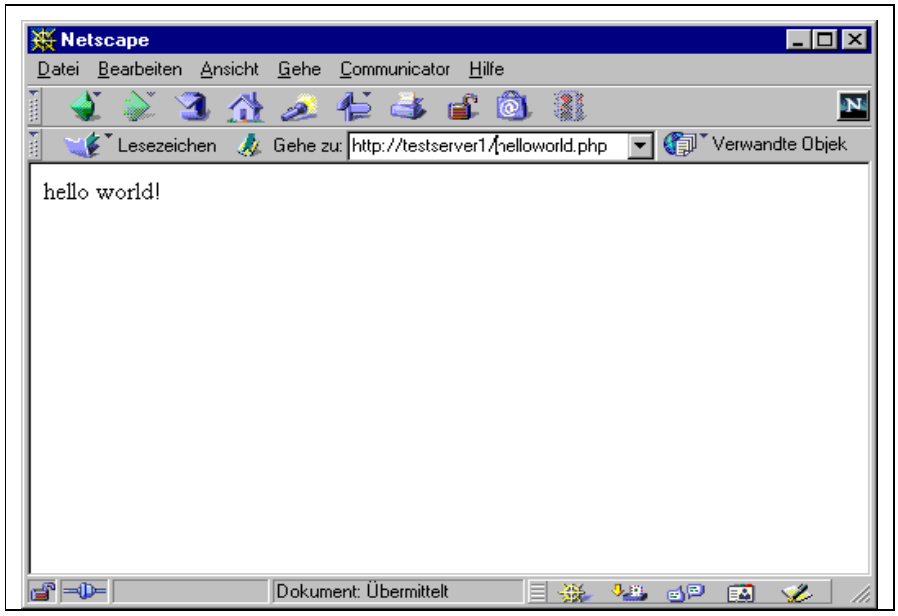

Abb. 4-3: helloworld.php Ausgabe im Browser

Die Verwendung einer server-seitig interpretierten Skriptsprache bietet natürlich noch weitere Vorteile. So können die Formulare sowohl zum Ausfüllen für den Bürger als auch zum Bearbeiten für den Sachbearbeiter benutzt werden. Es handelt sich dabei in beiden Fällen um die gleiche Datei, die jeweils nur in eine andere Umgebung eingebunden wird. Damit dies funktioniert, wurde folgende Unterteilung für die Formularskripte vorgenommen:

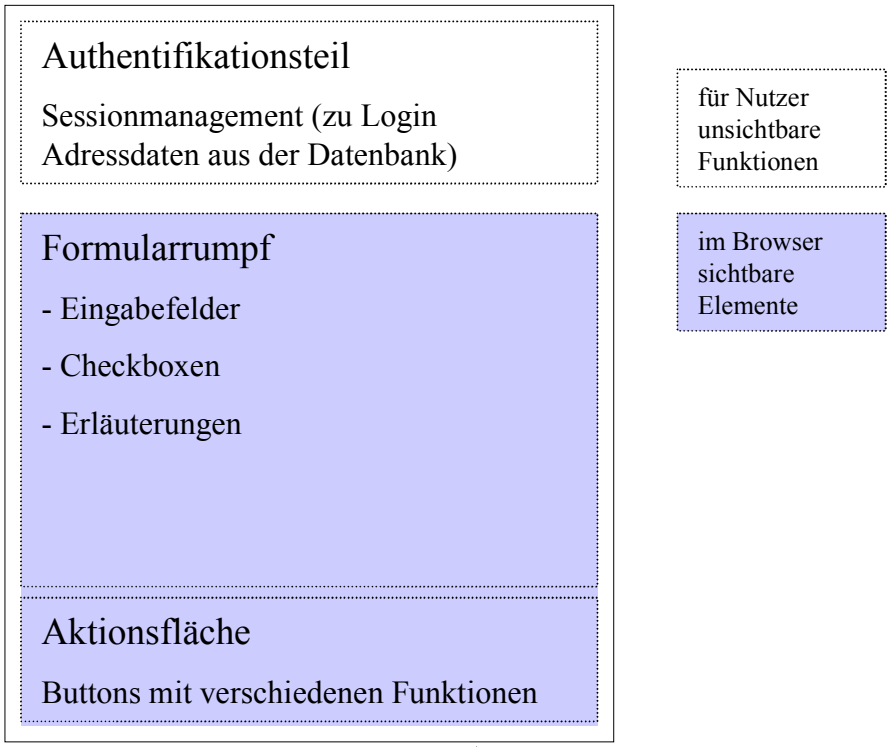

Abb. 4-4: Aufbau der Formularskripte (schematisch)

Der Authentifikationsteil ist für Bürger und Sachbearbeiter verschieden. Ruft ein Bürger das Formular auf, so wird er aufgefordert sich zu authentifizieren. Nach der erfolgreichen Anmeldung liest das Sessionmanagement die zum Bürger zugehörigen Informationen wie Wohnanschrift aus der Datenbank. Diese Daten werden dann im Formularrumpf automatisch ausgefüllt. Wird das Formular von einem Sachbearbeiter aus der Stadtverwaltung aufgerufen, wird auch nach dem Login und Passwort des Angestellten gefragt und zusätzlich werden die Informationen des Antragstellers aus der Datenbank gelesen. Die Unterscheidung, welcher Aufrufer welchen Authentifikationsteil bekommt, erfolgt dadurch, dass zwei Steuerungsskripte existieren, die die unterschiedlichen Teile zusammenführen. Für die Bürger ist dies in Abb. 4-5 dargestellt.

```
<?php
include "../../inc/buerger_session.inc";
if($error==""){
  include "feuer.inc";
  include "buerger_action.inc";
}else{
  echo ("Fehler in der Sessionverwaltung aufgetreten!<br>");
}
?>
</body>
</html>
```

Abb. 4-5: Aufbau des Steuerungsskripts (neuer_antrag_light.php)

Die Datei "buerger_session.inc" beinhaltet die Authentifikations- und Sessionmanagement-Funktionen für die Bürger. Das Steuerungsskript für die Sachbearbeiter würde dann wie in Abb. 4-6 dargestellt aussehen.

```
<?php
include "../../inc/verwaltung_session.inc";
if($error==""){
  include "feuer.inc";
  include "verwaltung_action.inc";
}else{
  echo ("Fehler in der Sessionverwaltung aufgetreten!<br>");
}
?>
</body>
</html>
```

Abb. 4-6: Aufbau des Steuerungsskripts (bearbeiten_antrag_light.php)

Der nächste Teil ist das eigentliche Formular, welches hier zur Vereinfachung nur "feuer.inc" genannt wurde. Hier werden die einzelnen Eingabefelder, Erläuterungen und sonstigen Elemente platziert, die für die Beantragung für den jeweiligen Vorfall benötigt werden. Die vollständigen Skripte (neuer_antrag.php und bearbeiten_antrag.php) erkennen die zugehörigen Formulare, die sie einbinden müssen, anhand eines übergebenen Parameters (dem Antragsnamen). Mit dessen Hilfe wird aus der Datenbank der Pfad zur Datei gesucht und diese eingebunden.

Damit ergeben sich große Vorteile für die Wartung der Anträge:

- Wenn ein neuer Antrag für die Bürger entworfen wurde, so steht dieser sofort ohne weiteren Aufwand den Sachbearbeitern für die Bearbeitung zur Verfügung - es handelt sich ja um ein und dieselbe Datei, welche nur in verschiedene Umgebungen eingebunden wird.
- Veränderungen an einem Antrag (Hinzufügen von weiteren Elementen) werden auch für die Bearbeiter sichtbar.

Als letzter Teil wird die Aktionsfläche eingebunden, welche die Formular-Buttons beinhaltet. Für die Bürger werden die Buttons "Löschen" und "Absenden" angegeben (Datei buerger_action.inc). Für die Sachbearbeiter werden weitere Elemente benötigt. Hierbei handelt es sich um die Buttons "speichern", "zurücksetzen" und "drucken", die Radio-Buttons "genehmigt", "abgelehnt" und "rückfragen" sowie ein Textfeld für Bemerkungen (vgl. Abb. 4-15).

4. 2 Die Speicherungskomponente

4. 2. 1 Produkt

Zur Speicherung der Antrags- und Steuerdaten wird ein DBMS verwendet. Das ausgewählte *relationale DBMS* (*RDBMS*) ist MySQL der schwedischen Firma MySQL AB. Wie schon in Abschnitt 2. 4. 2 erwähnt ist der große Vorteil dieses Produkts darin zu sehen, dass man für die meisten Einsätze keine Lizenzgebühren zahlen muss. Dies kommt den sowieso knappen Kassen einer Stadtverwaltung sehr entgegen. Weiterhin wird MySQL von verschiedenen Webserver-Erweiterungen unterstützt. Auch die geringen Ressourcen-Anforderungen sprechen für einen Einsatz im Low-Cost Bereich. MySQL nutzt die *GLP* (*GNU General Public License*), welche beschreibt wie die Software in verschiedenen Situationen genutzt werden darf. MySQL ist Open Source Software, was bedeutet, dass sie für jeden einsetzbar und leicht an die eigenen Anforderungen anpassbar ist. Neben der Linux-Variante gibt es noch weitere Versionen für andere Betriebssysteme (Solaris, SunOS, AIX, FreeBSD, Openbsd, HPUX, Microsoft Windows 9x/ NT und andere). Zwar gibt es neben MySQL noch ähnliche Produkte, wie zum Beispiel PostgreSQL [PSQL01], welche aber aus unterschiedlichen Gründen nicht gewählt wurden. Bei PostgreSQL war dies konkret die schlechtere Lauffähigkeit unter Microsoft Windows Betriebs-

systemen, da PostgreSQL nur unter einer cygwin-Emulation läuft.
Wie in Abschnitt 2. 4. 2 beschrieben, bietet MySQL keine direkte Transaktionsunterstützung. An einer einzigen Stelle im System würde diese Funktionalität benötigt. Dies wurde aber durch die Verwendung von LOCK TABLE umgangen. Dabei wird die Tabelle für einen kleinen Zeitraum gesperrt, die Operationen auf dieser Tabelle werden ausgeführt und die Tabelle wird danach gleich wieder freigegeben. Das Sperren der Tabelle ist hier vertretbar, da es sich um zwei kurze und einfache Operationen handelt, die schnell ausgeführt werden können. An anderen Stellen werden keine Transaktionen benötigt, weil jede Operation einzeln durchgeführt werden soll.

4. 2. 2 Datenbankschema

Ein Datenbankschema dient der abstrakten Darstellung eines Ausschnitts der Realität mit ihren Entitäten (Objekten) und deren Beziehungen untereinander. Es stellt einen Zwischenschritt dar, von dem aus die Abbildung in eine elektronisch verarbeitbare Form erleichtert wird.
Dabei gelten folgende Konventionen:

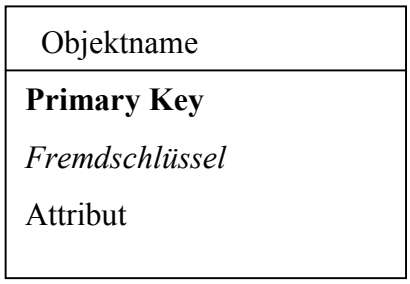

Abb. 4-7: Tabellensymbol

Dieses Symbol beschreibt eine Entität. Der Name wird im oberen Kasten angegeben. Der untere Kasten enthält die einzelnen Attribute. Diese können auch noch zusätzliche Informationen beinhalten. So werden Primärschlüssel (die Werte dieses Attributs müssen eindeutig sein) fett geschrieben, während die Fremdschlüssel (über diese Attribute werden Beziehungen zu anderen Objekten modelliert) kursiv wiedergegeben werden.
Die Abbildung 4-8 soll einen Überblick über die in die Datenbank zu integrierenden Objekte und deren Eigenschaften geben. Die Rechtecke stellen Objekte mit ihren Attributen dar, die

Verbindungen zwischen diesen stehen für die Beziehungen zwischen den Objekten.
Inhalte von Datenbanktabellen werden an verschiedenen Stellen im System benötigt. Das fängt beim Speichern der vom Bürger eingegebenen Daten an, führt über die automatische Benachrichtigung der zuständigen Mitarbeiter, die auch aus Datenbanktabellen ausgelesen werden, bis hin zur Recherche und zum Einpflegen neuer Tabellen beim Hinzufügen von neuen Anträgen.

Da in diesem System nur registrierte Bürger (Tabelle **Buerger**) Anträge stellen sollen, müssen der Name, die Adresse, eventuell eine E-Mail-Adresse und ein eindeutiges Login von jedem Bürger gespeichert werden. Die **Mitarbeiter** der Stadtverwaltung, die später einmal die Pflege und Bearbeitung der Anträge übernehmen sollen, müssen auch namentlich erfasst werden. Hier können auch gleich die Kontaktdaten (E-Mail, Telefon) mit angegeben werden. Damit ist es dann möglich, zu einem Antrag gleich den richtigen Ansprechpartner und dessen E-Mail-Adresse und Telefonnummer auszugeben, wenn es sich um einen Antrag handelt, für den nur ein Mitarbeiter zuständig sein sollte. Andernfalls werden die Kontaktdaten der zuständigen **Rolle** ausgegeben. Zu jeder Rolle gibt es eine E-Mail-Adresse und eine Telefonnummer. Die Rollenzuordung erfolgt in der Tabelle Mitarbeiter unter dem Fremdschlüssel Rolle. Es muss nicht ein einziger Mitarbeiter für einen Antrag zuständig sein, es können auch mehrere oder eine ganze Abteilung derselben Rolle zugeordnet werden. Dies ist bei Abwesenheit einer Person oder Antragsflut sehr hilfreich, da mehrere Mitarbeiter über neue Antragseingänge informiert werden können.
Jeder Antrag (zum Beispiel **Antrag_baumfaellung**) beinhaltet mehrere Felder, deren Inhalt für die Bearbeitung die zentrale Rolle spielen. Je nach Antrag kann die Anzahl dieser Felder variieren. Um den Überblick über die vorhandenen Anträge nicht zu verlieren, müssen diese an einer ausgezeichneten Stelle gespeichert werden (**Antragstabellen**). Hier soll auch gleich die zugehörige Rolle abgelegt werden.
Alle ankommenden Anträge werden im **Eingang** registriert. Das bedeutet, dass ihre Antragsnummer, die Antragsart (welcher Antrag) und das Eingangsdatum gespeichert werden. Um im Nachhinein feststellen zu können wer wann welchen Antrag bearbeitet hat und zur Feststellung, ob der Antrag schon abgeschlossen ist, werden alle Bearbeitungsvorgänge registriert (Tabelle **Bearbeitung**).

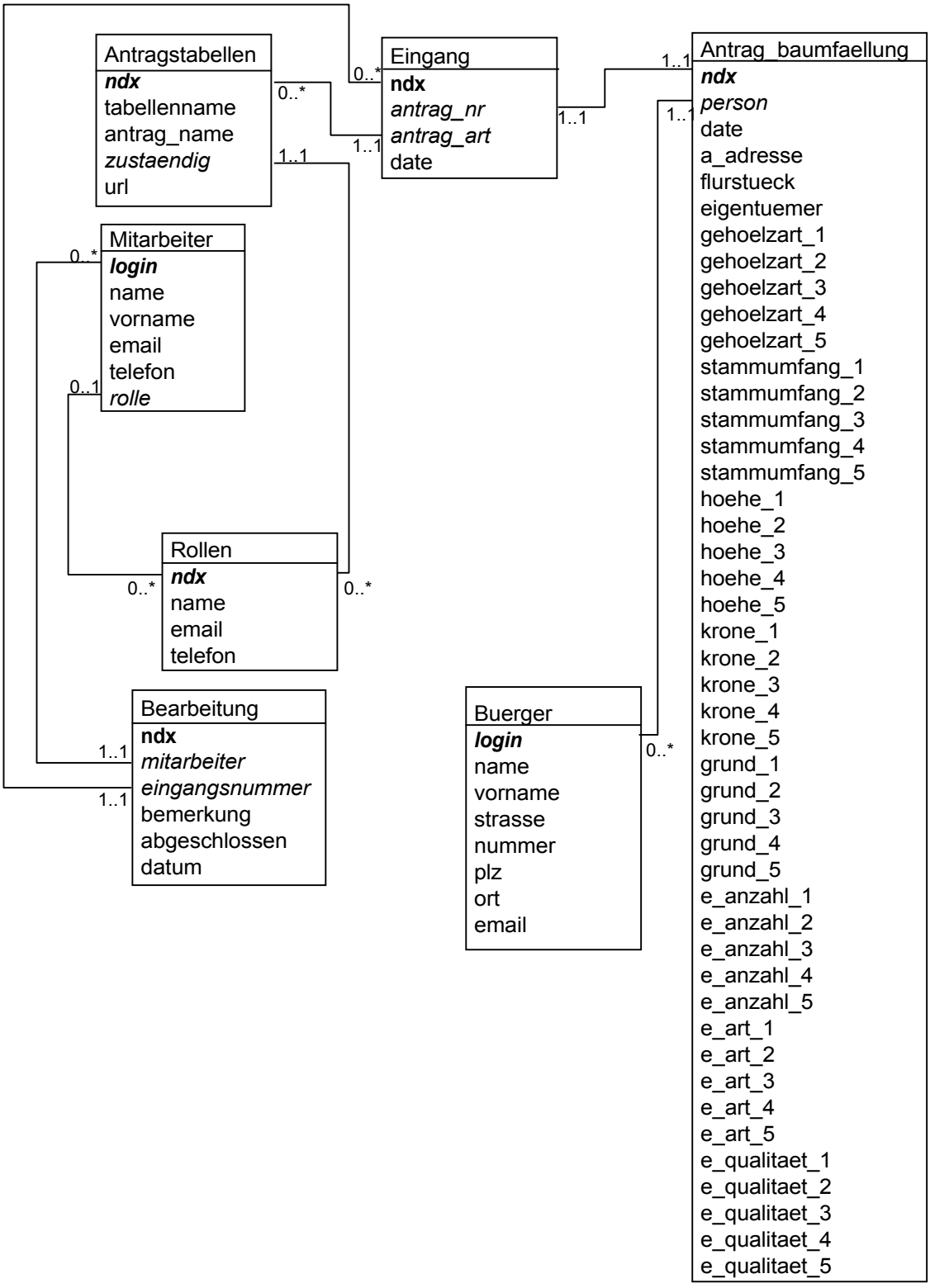

Abb. 4-8: Datenbankschema

Man kann zwischen zwei verschiedenen Tabellenarten in diesem System unterscheiden. Auf der einen Seite die, die für die Steuerung und die Funktionalität verantwortlich sind. Auf der anderen Seite finden sich die Tabellen wieder, welche die Daten der online gestellten Anträge beinhalten. Tabelle 4-1 ordnet die Tabellen des DB-Schemas den beiden Klassen zu.

Steuerung/Funktionalität	Daten
Mitarbeiter (beinhaltet die Daten der Mitarbeiter)	Buerger (beinhaltet die Daten der registrierten Bürger)
Rollen (definiert die einzelnen Rollen)	Antrag_baumfaellung und andere Antragstabellen
Antragstabellen (fasst alle vorhandenen Antragstabellen zusammen)	Eingang (beinhaltet eine Übersicht über alle eingegangenen Anträge)
	Bearbeitung (dient der Speicherung der abgeschlossenen Bearbeitungsvorgänge)

Tabelle 4-1: Tabellen und deren Zuordnung

4. 2. 3 Neue Antragstabelle

Für jeden in das System integrierten Antrag wird eine eigene Datentabelle erstellt. Dies kann automatisch über ein Skript erfolgen, welches natürlich nicht die optimale Lösung bieten kann, weil das Anlegen der Tabelle nicht trivial ist. Dieses Skript (**mk_antrag.php)** soll dennoch einmal angesprochen werden, da es auch Installateuren ohne Datenbankkenntnisse erlaubt neue Anträge in das System zu integrieren. An dieses Skript kann man das neue Formular folgendermaßen schicken:
Man ersetzt im FORM-tag den Action-Wert

```
<form action=mk_antrag.php method=post>
```

und öffnet das Formular einmal mit dem Browser. Dann klickt man einfach einmal auf Absenden und das Skript sammelt nun alle übergebenen Eingabefelder und legt anhand derer die Tabelle an, falls diese noch nicht existiert. Damit dies funktioniert, müssen verschiedene Bedingungen erfüllt sein, die der Entwickler des HTML-Formulars berücksichtigen muss. Zum

einen muss es ein sogenanntes verborgenes Input-Feld geben, welches den Namen "tabellenname" hat. Als Inhalt (value) muss der Name der Tabelle angegeben werden, die für diesen Antrag genutzt werden soll. Weiterhin muss es ein Feld mit dem Namen "antragsname" geben. Dessen Inhalt wird in der Tabelle Antragstabellen beim Anlegen der Datentabelle eingefügt. Ein Beispielfragment eines HTML-Formulars soll dies veranschaulichen:

```
<html>
[...]
<body>
<form action="mk_antrag.php" method="post">
<input type="hidden" name="tabellenname" value="Antrag_baumfaellung">
<input type="text" name="antragsname" value="Antrag auf Baumfällung">
[...]
</form>
</body>
</html>
```

Nichtsdestotrotz ist es aber in jedem Falle vorzuziehen die neuen Antragstabellen von einem Datenbankadministrator nach Analyse des Formulars anlegen zu lassen. Dabei sollte vor allem auf die zu erwartenden Eingabelängen der einzelnen Felder geachtet werden.

4. 3 Administrationskomponente

Damit die Benachrichtigungen über neue Antragsdaten auch die richtigen Mitarbeiter erhalten, muss die Zuständigkeit der Mitarbeiter zu den Anträgen geklärt werden. Dazu bedarf es eines Administrationstools, welches ausgewählten Personen innerhalb der Stadtverwaltung erlaubt, dem System Mitarbeitergruppen als Bearbeiter für bestimmte Anträge zu benennen. Wie aus dem Datenbankschema hervorgeht, werden nicht einzelne Mitarbeiter zu Anträgen zugeordnet sondern Rollen. Mitarbeiter sind Mitglied einer oder mehrerer Rollen. Eine Rolle ist in diesem Fall vergleichbar mit einem Amt. Denn auch in einem Amt sollten dessen Mitarbeiter kompetent sein Anträge an das Amt zu bearbeiten. Es ist daher auch sinnvoll die Rollennamen aus den Namen der Ämter abzuleiten. Damit ist später eine leichtere und übersichtlichere Zuordnung möglich. Sind die ersten Rollen erstellt, so kann mit der Aufnahme der Mitarbeiter und deren Rollenzuordnung begonnen (vgl. Abb. 4-13) werden. Jeder Mitarbeiter erhält ein Login und ein Passwort, mit denen er sich am System authentifizieren muss, bevor er geschützte Bereiche be-

tritt. Dies ist zum Beispiel für die Bearbeitung von Antragsdaten notwendig.

4. 4 Präsentationskomponente

Für die Darstellung der Daten und Bereitstellung der Funktionen wird ein Browser benutzt, welcher die von PHP dynamisch erzeugten HTML-Seiten anzeigt. Im vorliegenden Fall wird die Benachrichtigung der Mitarbeiter durch E-Mail realisiert. Dies setzt natürlich einen installierten E-Mail-Client auf dem Mitarbeiterrechner und einen Mailserver auf dem Web-Server voraus. Die Benachrichtigungs-E-Mails enthalten keinerlei personenbezogene Daten der Antragsteller, nur einen Link, der die Mitarbeiter zum zu bearbeitenden Datensatz führt (vgl. Abb. 4-9).

X-Mailer:. QUALCOMM Windows Eudora Version 4.3.2
Date: Fri, 14 Dec 2001 10:00:00 + 0200

To: ordnung@sv.de
From: Dataserve
Subject: neuer Antrag auf Baumfaellung

Sehr geehrter Herr Mustermann,

es liegen neue Antragsdaten in Ihrem Zustaendigkeitsbereich vor.
Bitte klicken Sie hier.

Diese Nachricht wurde automatisch erstellt.
Bei Problemen wenden Sie sich bitte an admin@sv.de

Abb. 4-9: Benachrichtigungs-E-Mail

4. 5 Steuerungskomponente

4. 5. 1 Aufbau

Die Aufgabe der Steuerung des gesamten Systems kann entweder durch ein kompiliertes Programm oder aber durch ein (Web-) Interface unter Einbeziehung von Steuerungsdaten aus einer Datenbank oder einer Parameterdatei erfolgen. Ein Programm hat den Vorteil, dass es ein einfaches und kompaktes Produkt sein würde, welches aber nur vorhersehbare Fälle berücksichtigt und das Hinzufügen von neuen Abläufen oder deren Änderung kaum ermöglicht. Weiterhin muss man sich vor der Programmierung auf ein Zielsystem (Computertyp, Betriebssystem) festlegen und dieses dann beibehalten. Anders hingegen beim Einsatz eines (Web-)Interfaces, bei dem die Steuerung durch die Auswertung von Daten mittels Skripten oder durch Java-Applets erfolgt. Durch die Forderungen, welche durch die heterogene Hardware an das System gestellt werden, ist es sinnvoll auf ein kompiliertes Programm zu verzichten und stattdessen eine Web-Applikation zu entwerfen.
Die Steuerungskomponente besteht in dem in der vorliegenden Arbeit zu entwickelnden System aus der Datenbank, welche Daten zur Regelung des Gesamtablaufs beinhaltet und den PHP-Skripten, welche diese Daten auslesen und entsprechende Aktionen ausführen. Indirekt gehört natürlich auch der Web-Server dazu, da er die Abarbeitung und Interpretation der Skripte erst ermöglicht.

4. 5. 2 Formulargestaltung

Die Funktionsweise der Steuerungskomponente soll in diesem Abschnitt anhand eines konkreten Beispiels zum kompletten Ablauf vom Erstellen eines neuen Antrages über dessen Nutzung durch Bürger bis zur Bearbeitung durch Mitarbeiter der Stadtverwaltung dargestellt werden. Es wird dabei auf die benötigten Daten und die jeweiligen Skripte eingegangen.
Das Problem: Es soll ein Antrag auf Baumfällung für die Bürger online angeboten werden. Diesen Antrag gibt es bis jetzt nur in Papierform. Auf Grundlage dessen wird mit Hilfe eines HTML-Editors ein HTML-Formular erstellt, welches die Eingabe der benötigten Daten ermöglicht (siehe Abb. 4-10: Formular Baumfällung).
Nach der graphischen Umsetzung geht es nun um den Inhalt. Jedes Eingabefeld muss einen ein-

deutigen Namen aufweisen, welcher nicht durch Sonderzeichen oder Leerzeichen getrennt sein darf.

Zusätzlich müssen noch unsichtbare Input-Felder eingebaut werden, die zur Steuerung der Datenbank notwendig sind (vgl. Abschnitt 4. 2. 3). Dies sind die Felder "antragsname" und "tabellenname". Anhand des Inhalts des Feldes "tabellenname" wird vom System später eine Tabelle gleichen Namens angelegt. Der Inhalt des Feldes "antragsname" dient der Erläuterung der Tabelle - in diesem Fall "Antrag auf Baumfällung".

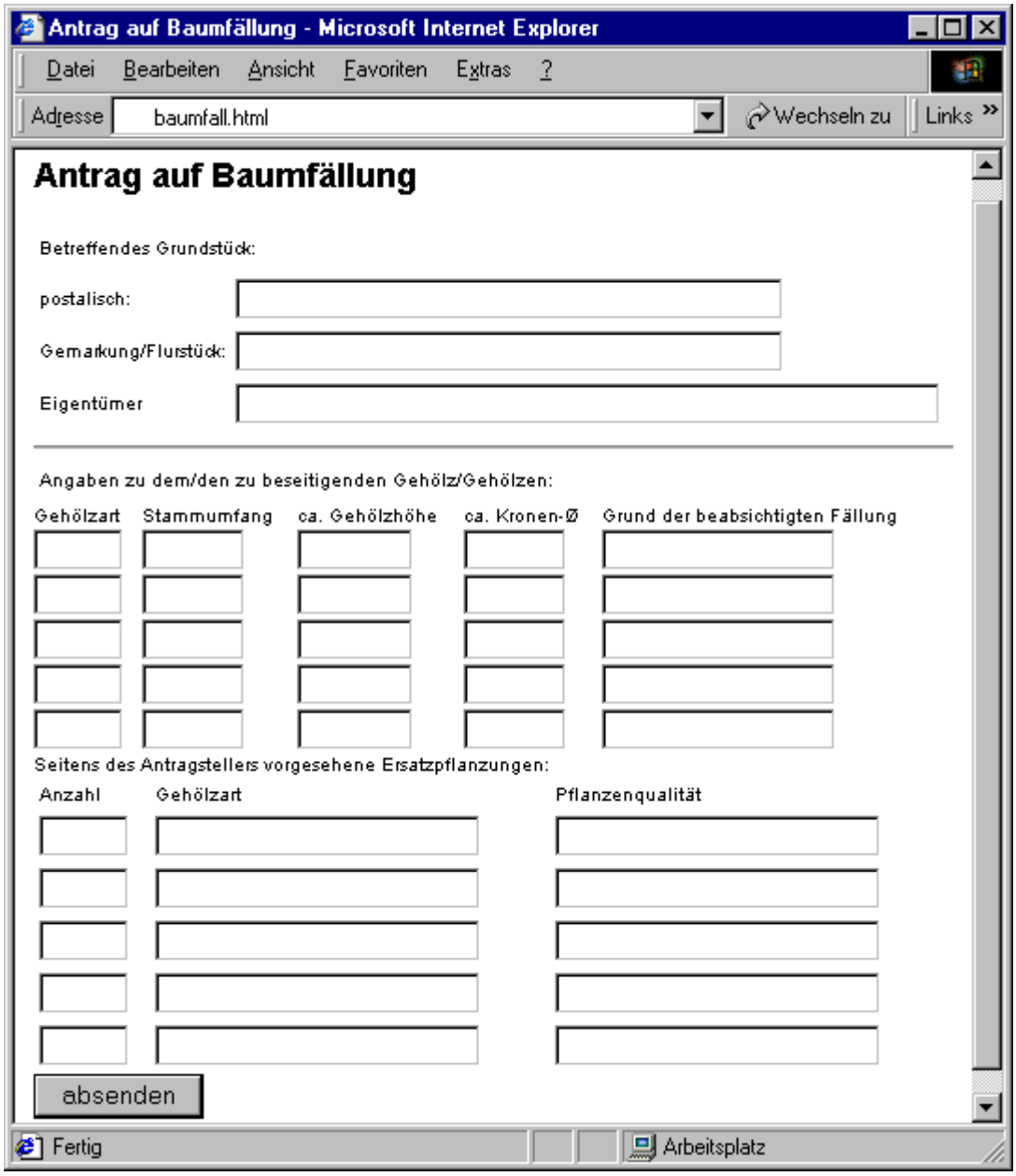

Abb. 4-10: Formular Baumfällung

4. 5. 3 Stellung eines neuen Antrags

Der Bürger klickt im Online-Auftritt der Stadt den Link zum Antrag an und wird nach seinem Login/Passwort gefragt. Diese Eingabe wird vom System mit Hilfe der Tabelle Personen überprüft und bei Übereinstimmung wird der Antrag dem Bürger zum Ausfüllen angezeigt. Durch Javascript-Funktionen kann hier schon eine erste Plausibilitätsprüfung der eingegebenen Daten erfolgen. Dies ist jedoch von Antrag zu Antrag unterschiedlich und muss vom jeweiligen HTML-Formular-Designer bei der Entwicklung der HTML-Datei berücksichtigt (das heißt programmiert) werden.
Sind alle Daten korrekt eingegeben, so drückt der Bürger den "absenden"-Knopf. Daraufhin werden die Daten an ein Skript auf dem Server übermittelt. Dort werden die einzelnen Inhalte der Eingabefelder des Formulars in die Tabelle gespeichert, die im Eingabefeld mit dem Namen "tabellenname" bezeichnet wurde (wenn die Tabelle schon vorhanden ist, vgl. Abschnitt 4. 2. 3).
An dieser Stelle startet eine Überprüfung, welche "Dauerantragssteller" überführt und dem Antragsteller eine Information zur Erinnerung ausgibt, dass er diesen Antrag schon vor kurzer Zeit gestellt hat und, falls dieser noch nicht abgeschlossen ist, er doch noch etwas Geduld zeigen möchte.
Weiterhin sollte, wenn keine Plausibilitätsprüfung vor dem Absenden des Antrags durchgeführt wurde, diese spätestens hier (vor Speicherung in die Datenbank) erfolgen. Damit wird verhindert, dass unvollständige oder sinnlose Anträge das System und die Mitarbeiter unnötig belasten. Sollte es hingegen keine Probleme geben und die Daten ordnungsgemäß eingegeben worden sein, dann wird ein neuer Eintrag in der Tabelle **Eingang** generiert, der die Antragsnummer, das Eingangsdatum und den Antragsnamen beinhaltet.
Danach benachrichtigt das Skript die zuständigen Mitarbeiter per E-Mail und informiert den Antragsteller über den erfolgreichen Eingang seiner Daten. Sollte ein Fehler aufgetreten sein, so wird der Bürger darüber in Kenntnis gesetzt und zur erneuten Eingabe eventuell zu einem späteren Zeitpunkt aufgefordert.

4. 5. 4 Bearbeitung des Antrags

Wie in Abschnitt 4. 4 beschrieben werden die zuständigen Sachbearbeiter per E-Mail über den Eingang eines neuen Antrags benachrichtigt.

Der Mitarbeiter braucht nun nur noch auf den Link zu klicken und erhält die vom Antragsteller eingegebenen Daten aus der Antragstabelle plus die persönlichen Daten der Person aus der Tabelle Bürger. Es haben aber nun alle Mitarbeiter diese Benachrichtigungs-E-Mail bekommen, die in den Zuständigkeitsbereich des Antrags fallen (aus Tabellen Antragstabellen, Rollen, Rollenzuordnung). Da HTTP keine Transaktionen unterstützt, kann es zu dem Phänomen kommen, dass zwei Sachbearbeiter den gleichen Antrag bearbeiten. Um dies zu verhindern, gibt es folgenden Mechanismus:
Der erste Mitarbeiter, der den Link in der E-Mail anklickt, erzeugt automatisch den ersten Eintrag für diesen Antrag in der **Bearbeitung**stabelle. Dieser beinhaltet neben dem Datum und der Eingangsnummer des Antrags auch das Mitarbeiter-Login, ein Bemerkungsfeld und ein Feld zur Kennzeichnung, ob der Antrag abgeschlossen ist. Alle weiteren Mitarbeiter, die danach den Link anklicken, bekommen nun eine Information, dass dieser Antrag von Mitarbeiter xy bearbeitet wird beziehungsweise wurde, wenn der Antrag schon abgeschlossen ist.
Für den Fall, dass der Antrag durch verschiedene Ämter bearbeitet werden muss, können Mitarbeiter des anderen Amtes nun unabhängig davon, ob sich schon ein Mitarbeiter (eines anderen Amtes) mit den Antrag befasst, diesen bearbeiten. Dazu wird wieder ein Eintrag in der **Bearbeitung**stabelle erzeugt.
Im Feld Bemerkung können Hinweise über den Zustand der Bearbeitung hinterlegt werden, die eventuell auch die Mitabeiter des zweiten beteiligten Amtes interessieren. Sollte ein Mitabeiter mit diesem Antrag nichts anfangen können, was sicherlich auch mal vorkommen darf, so wird sein Eintrag aus der **Bearbeitung**stabelle wieder entfernt. Damit können sich nun andere um diesen Antrag kümmern.
Der Eintrag in der Eingangstabelle ist keine Sperre, er dient lediglich als Hinweis für die Mitarbeiter, dass sich schon jemand um diese Daten bemüht und soll damit die doppelte Bearbeitung verhindern. Sollte es notwendig sein, dass andere Ämter oder Abteilungen auch Zugriff auf diese Daten benötigen, so ist dies natürlich möglich.

4. 5. 5 Abschluss eines Antrags

Wenn der Antrag durch einen Mitarbeiter genehmigt oder abgewiesen wurde, wird das Flag "abgeschlossen" in der Tabelle Bearbeitung gesetzt und der Abschlusstext wird in der Spalte "bemerkung" festgehalten. Daraufhin wird eine E-Mail an den Antragsteller mit dem Inhalt der

Bemerkungsspalte verschickt.
Dadurch dass eine E-Mail noch nicht die hohe Beweisfähigkeit besitzt wie beschriebenes Papier, gibt es eine Funktion in diesem System, welche alle abgeschlossenen Anträge anzeigen kann, damit diese gleich gedruckt und dann per Post an den Antragsteller verschickt werden können. Dieses Problem des Medienbruches sollte demnächst aufgehoben werden, wenn die digitale Signatur eine breite Unterstützung in der Bevölkerung erfährt (vgl. Abschnitt 3. 5).

4. 6 Die Recherchekomponente

4. 6. 1 Eigenschaften

Durch den Einsatz des Systems können auch intensive Recherchen auf dem Datenbestand durchgeführt werden. Diese reichen vom einfachen Suchen von Anträgen zu einer gegebenen Person über die Auswertung wieviele Anträge pro Monat gestellt werden bis hin zur detaillierten Suche nach bestimmten Eigenschaften, die in Anträgen angegeben wurden. Auch die benötigte Bearbeitungszeit jedes einzelnen Antrags ist recherchierbar. Möglich macht dies die automatische Erfassung der Bearbeitungsschritte.
Durch die Variabilität des Systems und den daraus resultierenden unvorhersehbar zum Einsatz kommenden Anträgen ist es nicht möglich ein statisches Recherchetool zu entwerfen, welches alle möglichen Kombinationen berücksichtigt. Folgendes einfache Beispiel soll dies verdeutlichen: Ein Mitarbeiter möchte herausfinden, für welche Straßen im letzten Jahr eine Genehmigung für die Sondernutzung von Verkehrsraum beantragt wurde. Eine Lösung dazu sieht folgendermaßen aus:
Er wählt im Recherchetool aus, welche Anträge für seine Anfrage untersucht werden sollen. Hier werden es die „Anträge auf Sondernutzung von Verkehrsraum“ sein. Um die Ergebnisse weiter einzuschränken wählt er nun die Felder aus, die ihn besonders interessieren. Das wäre das Feld „Ort der Sondernutzung“. Nun könnte er den Namen der gesuchten Straße eingeben und erhält eine Liste von Anträgen, die eine solche Sondernutzung beantragt haben.
Das Problem dabei: Niemand weiß bei der Installation geschweige denn bei der Implementation, welche relevanten Felder es später einmal geben wird. Es ist nicht einmal klar wie der Name der Tabelle sein wird. Folglich muss ein dynamisches Tool entwickelt werden, welches abhän-

gig davon ist, welche Tabellen und damit Anträge im jeweiligen System integriert wurden.

4. 6. 2 Funktionsweise

Für die Realisierung spielen die „Steuerungs"-Tabellen (vgl. Abschnitt 4. 2. 3) eine wichtige Rolle. Vor allem die Tabelle „Antragstabellen" dient hier als Grundlage. Die einzelnen in einer Antragstabelle zur Verfügung stehenden Felder müssen jedoch aus der Datenbank selbst extrahiert werden. Das zu entwickelnde Tool muss nach folgendem Schema (Abb. 4-11) funktionieren:

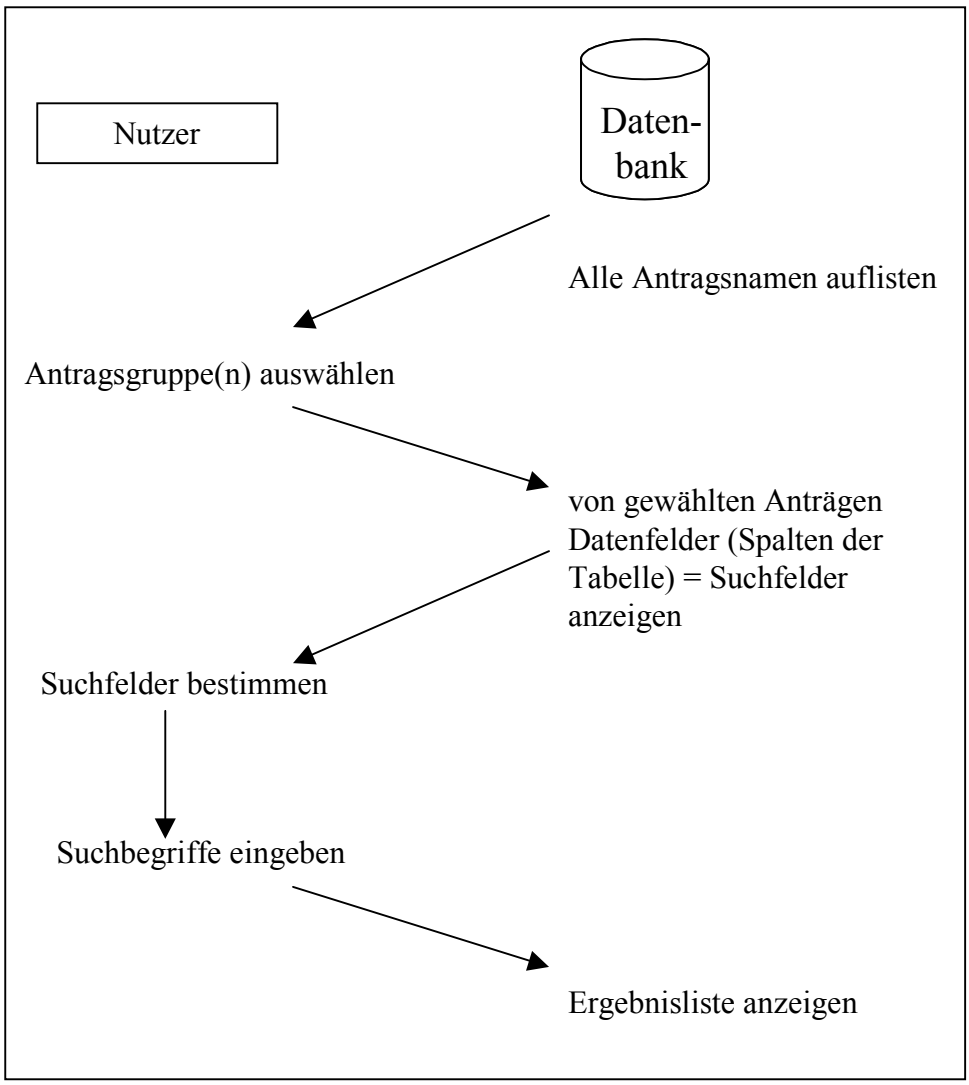

Abb. 4-11: Arbeitsschritte Recherchekomponente

4. 7 Die Sicherheitskomponente

Damit nur berechtigte Personen Zugang zu den gespeicherten Daten und der Administration des Systems erhalten, wurden in der Tabelle "Mitarbeiter" in der Datenbank für jede Person, die mit dem System zu tun hat, ein Login und ein Passwort vergeben. Anhand der Zuordnung zu einer bestimmten Rolle entscheidet die Steuerungskomponente, welchen Ausschnitt der Datenbank dieser Nutzer bearbeiten darf. Wenn er zum Beispiel der Rolle "Ordnungsamt" zugeordnet wurde, dann kann er nur solche Anträge bearbeiten, welche in der Spalte "zustaendig" der Tabelle "Antragstabellen" den Wert "Ordnungsamt" enthalten.
Für die Administratoren, welche den organisatorischen Ablauf in der Datenbank definieren (z.B. Rollenzuordnungen, Nutzer anlegen) gibt es eine extra Rolle (system), damit diese Nutzer von den übrigen Mitarbeitern unterschieden werden können. Durch diese Rolle, die fest in den Skripten vorgegeben ist, können für diese Nutzer von den Skripten andere Funktionen zur Verfügung gestellt werden.
Damit Außenstehende erst gar nicht in die Versuchung kommen sich am System anzumelden, wurde noch ein Schutz der Skriptverzeichnisse auf dem Webserver installiert. Dabei wird die .htaccess Datei (vgl. [EIL98]) im Verzeichnis, in dem die Skripte zur Bearbeitung und Administration liegen, genutzt. In der Datei access.conf im Verzeichnis www/etc/apache/ musste die globale Deaktivierung der .htaccess Dateien geändert werden, so dass in jedem Fall bei jedem Zugriff auf ein Verzeichnis des Webservers die .htaccess Datei beachtet wird (falls vorhanden). Dies geschieht durch folgenden Eintrag:
Datei *access.conf*:

```
< Directory />
AllowOverride All
</Directory>
```

Im Verzeichnis der Skripte wurde eine Datei .htaccess mit folgendem Inhalt angelegt:
Datei *.htaccess*:

```
<Files *.php>
order deny, allow
deny from all
allow from 138.201
</Files>
```

Diese Anweisung veranlasst den Webserver bei jedem Zugriff auf eine Datei mit der Endung .php in diesem Verzeichnis zu überprüfen, woher die Anfrage nach der Datei kam. Wurde die Anfrage von einem Rechner außerhalb des IP-Bereichs von 138.201 und dessen Unternetz gestellt, so gibt der Webserver nur eine Fehlermeldung an den Client zurück, dass er keine Berechtigung habe, auf diese Datei zuzugreifen. Alle anderen Rechner, welche eine IP im Bereich von 138.201.0.0 bis 138.201.255.255 (Netz in der Stadtverwaltung) aufweisen, dürfen auf diese Dateien zugreifen. "order deny, allow" gibt an, dass zuerst einmal alle Zugriffe auf die Dateien von allen Rechnern verboten sind bis auf die Ausnahmen welche unter "allow from" angegeben sind. Einen völligen Schutz gegen gezielte Angriffe bietet diese Maßnahme zwar noch nicht, aber für den Entwurf und die Implementierung eines Firewall-Systems müsste man die Gegebenheiten direkt vor Ort aufnehmen und anhand derer entscheiden, welches System mit welchen Komponenten zum Einsatz kommen soll.

4. 8 Test der Implementierung

Die oben beschriebene Implementierung wurde wie folgt getestet.
Auf einem Windows 2000 Server wurde der Apache Webserver mit PHP Erweiterung, das MySQL DBMS und ein E-Mail-Client (Microsoft Outlook-Express) installiert. Mittels des Skripts createDBObjects.sql wurden die Datenbankobjekte angelegt. Die PHP-Skripte wurden in das htdocs-Verzeichnis in die entsprechenden Unterverzeichnisse kopiert.
Die "Homepage" der Stadtverwaltungsangestellten (vgl. Abb. 4-12), über die das System administriert werden kann, wird im Browser mit http://127.0.0.1/index.html aufgerufen.
Um die Funktionstüchtigkeit des Systems zu testen werden zuerst Rollen, Mitarbeiter (s. Abb. 4-13) und Bürger über vorgefertigte Masken angelegt und in der Datenbank gespeichert. Danach werden den Mitarbeitern Rollen zugewiesen und ein Bürger aktiviert. Ein Antragsformular in HTML wird erzeugt, auf das bei der Antragsintegration (vgl. Abb. 4-14) verwiesen wird. Ebenfalls in dieser Maske werden ein Name für die Tabelle vergeben, in der die Antragsdaten später gespeichert werden, und die zuständige Rolle angegeben, die für die Bearbeitung der Antragsdaten zuständig ist.

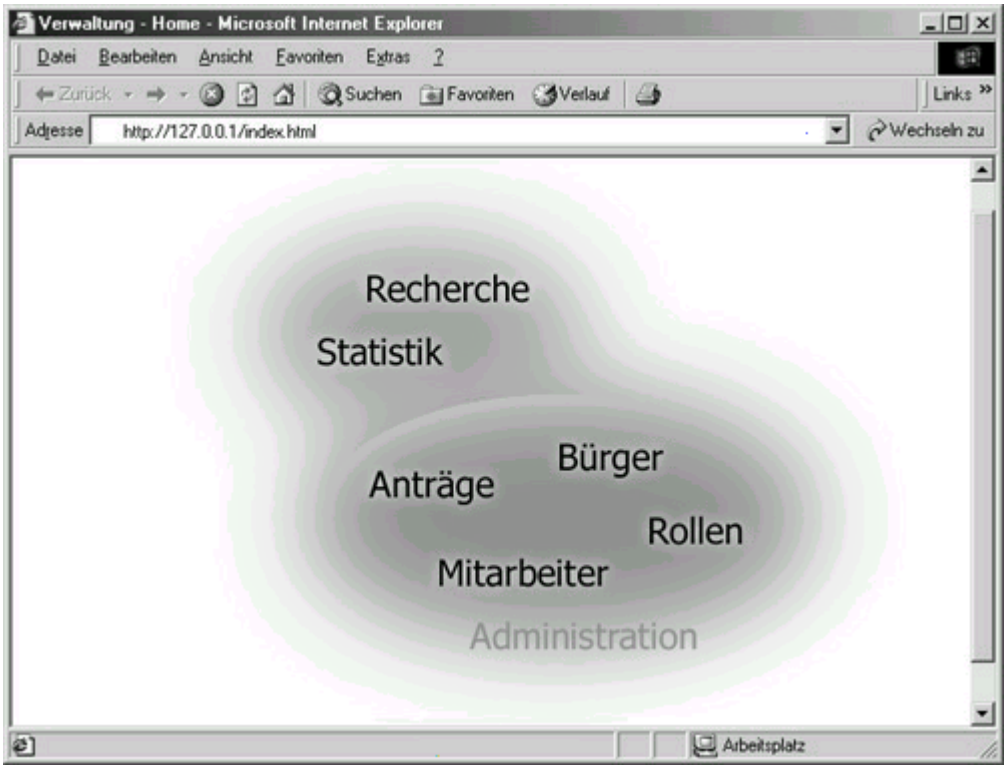

Abb. 4-12: Eingangsseite Stadtverwaltung

Als nächstes wird die Bürgerseite mit der Möglichkeit zur Eingabe der Daten für einen Antrag geöffnet. Erst nachdem das Login und das korrekte Passwort des Bürgers eingegeben wurden, erscheint das Antragsformular. Nachdem alle Daten eingegeben wurden, erscheint eine Bestätigung, dass die Antragsdaten gespeichert wurden, und die E-Mail-Adresse der Rolle, die für die Bearbeitung des Antrags zuständig ist.

Der zuständige Mitarbeiter erhält über den Posteingang der Rolle eine Mail mit einem Link zu den neuen Antragsdaten. Nach dem Klicken auf selbigen erscheint eine Aufforderung zur Eingabe des Logins und des Passworts des Sachbearbeiters. Erst nach korrekter Eingabe werden die Eingaben des Bürgers, die er bei der Antragstellung gemacht hat, angezeigt. Der Sachbearbeiter kann nun Vermerke zu diesen Daten machen und den Antrag in der Form ausdrucken, wie ihn der Bürger ausgefüllt hat (vgl. Abb. 4-15). Im Menüpunkt Recherche auf der Eingangsseite der Stadtverwaltung (Abb. 4-12) kann man den Antrag des Bürgers und dessen Bearbeitungszustand ansehen. Es ist vermerkt, welcher Sachbearbeiter wann diese Antragsdaten bearbeitet hat. Unter dem Menüpunkt Statistik ist neben dem Namen des Antrags die Zahl der bereits gestellten Anträge vermerkt.

Abb. 4-13: Maske zum Anlegen eines neuen Mitarbeiters

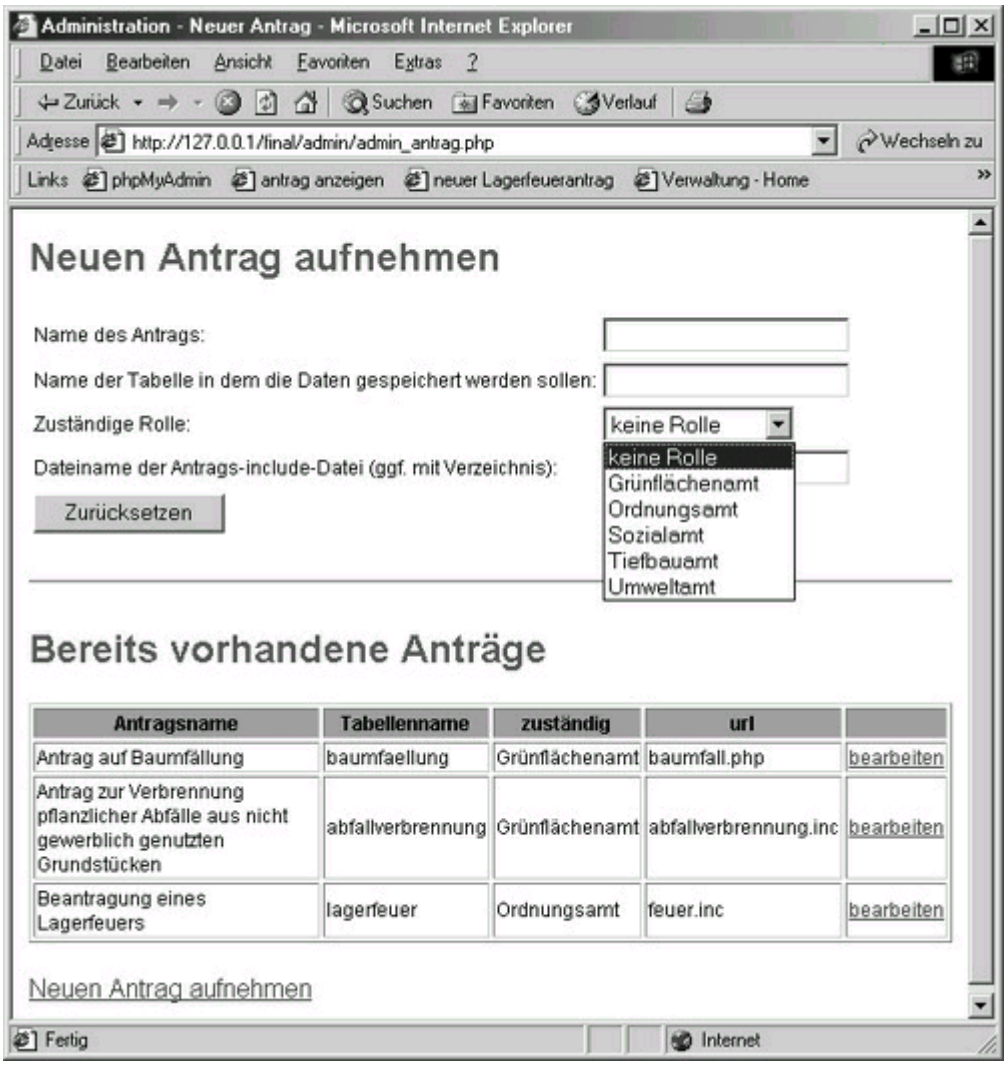

Abb. 4-14: Maske zum Anlegen eines Antrags

Damit wurde die Verwendbarkeit des entwickelten Systems für die wesentlichen Prozesse wie Verwaltung von Mitarbeitern, Bürgern und Anträgen, die Stellung von Anträgen durch Bürger, die Bearbeitung der Anträge in der Stadtverwaltung sowie die Recherche auf den Anträgen gezeigt. Außerdem wurde durch die Verwendung bereits vorhandener oder kostenlos zu beziehender Hard- und Software die Vorgabe erfüllt, das System mit möglichst geringem finanziellen Aufwand zu realisieren.

Abb. 4-15: Maske zur Bearbeitung von Antragsdaten

5 Zusammenfassung und Ausblick

5. 1 Zusammenfassung

In der vorliegenden Arbeit wurde ein System zum web-basierten Management von Daten aus Online-Anträgen von Bürgern an die Verwaltung konzipiert, welches die folgenden Aufgaben erfüllt:

1. Speicherung der vom Bürger im Online-Antrag eingegebenen Daten,
2. Steuerung und Protokollierung der Antragsbearbeitung,
3. Recherche auf den gespeicherten Anträgen,
4. Rollenmodell zur Regelung der Zuständigkeiten der Mitarbeiter für bestimmte Anträge,
5. Erweiterungsmöglichkeit durch Integration neuer Anträge und Kontrollmechanismen zur Vermeidung unbefugter Zugriffe.

Um diese Aufgaben zu realisieren wurde das System aus mehreren Komponenten aufgebaut. Dies sind im einzelnen:

- die Speicherungskomponente für die Ablage und Verwaltung der vom Bürger eingegebenen Daten (Aufgabe 1),
- die Steuerungskomponente für die Zuweisung der eingegangenen Anträge an die dafür zuständigen Mitarbeiter, wobei diese Komponente das Rollenmodell benutzt, um die passenden Bearbeiter für einen Antrag zu finden (Aufgaben 2 und 4),
- die Administrationskomponente zur Verwaltung der vorhandenen Anträge, zur Integration neuer Anträge und zur Festlegung der Rollen der Mitarbeiter (Aufgaben 4 und 5),
- die Präsentationskomponente, die für einheitliche Bearbeitungsoberflächen sowohl für die Sachbearbeiter als auch für den Administrator sorgt,
- die Recherchekomponente zur Suche auf den gespeicherten Anträgen (Aufgabe 3),

- die Sicherheitskomponente, die unter anderem für die Verwaltung von Zugriffsrechten für gespeicherte Anträge und für die Verhinderung unberechtigter Zugriffe verantwortlich ist (Aufgabe 5).

Bei der Implementierung war zu berücksichtigen, dass eine möglichst kostengünstige Realisierung erfolgen sollte. Das System steht demzufolge auf zwei Grundpfeilern, die häufig schon in der Verwaltung vorhanden sind oder mit geringem finanziellen Aufwand beschafft werden können. Erstens dem Webserver, welcher für die Ausführung von PHP-Skripten und damit die dynamische Generierung der HTML-Seiten zuständig ist, welche den Bürgern beim Stellen von Anträgen sowie den Mitarbeitern bei der Bearbeitung von Anträgen oder bei der Administration des Systems in ihren Browsern angezeigt werden. Zweitens dem Datenbank-Management-System, welches neben der reinen Speicherung der Antragsdaten zum Beispiel auch für die Zuordnung der Mitarbeiter zu bestimmten Anträgen, die Benachrichtigung von Mitarbeitern über neu eingegangene Anträge oder die Verwaltung von Zugriffsrechten zuständig ist.

5. 2 Ausblick

Eine nächste Version sollte für die Integration neuer Anträge eine verbesserte automatische Tabellenerzeugung beinhalten, da die bisherige Lösung recht ineffizient arbeitet. Sie geht zum Beispiel von festen Spaltengrößen für Eingabefelder aus und damit sehr verschwenderisch mit den Ressourcen des DBMS um. Abhilfe schafft hier eine Funktion, welche die neuen Anträge, die im HTML-Format vorliegen, nach bestimmten HTML-Tags durchsucht und anhand der Angaben in ihnen die Struktur der Tabelle aufbaut. Beispiel: in einem Formular kommt ein Eingabefeld für den Namen des Antragstellers vor:

```
[...]
<input name="a_name„ size=30 maxlength=30>
[...]
```

Dann kann anhand der Angabe „maxlength=30“ abgelesen werden, dass niemals mehr als 30 Zeichen übermittelt werden und somit die Länge der Spalte in der Tabelle nicht größer als 30 sein sollte.
Die Sicherheit sollte im praktischen Einsatz von einer Firewall ausgehen, die auch hartnäckigen

Angriffsversuchen standhält. Schließlich sind die in der Datenbank gespeicherten und abrufbaren Informationen nicht für jedermann bestimmt. Es ist daher auch darauf zu achten, dass auch innerhalb der Stadtverwaltung sich kein unberechtigter Mitarbeiter Zugriff auf die Daten verschaffen kann.

Durch die Einführung der digitalen Signatur können später auch Anträge mit rechtsverbindlicher Unterschrift von den Bürgern online gestellt werden. Damit dies geschehen kann, müssen die dafür notwendigen Grundlagen geschaffen werden (z.B. Serverzertifikat, siehe auch [Reg01]).

Literaturverzeichnis

[ADO01] Adobe Systems Website, http://www.adobe.de

[AHS01] Allaire Homesite Homepage, http://www.allaire.com

[APA01] Apache Website, http://www.apache.org

[BAU96] Baumgarten, B. (1996): Petri-Netze: Grundlagen und Anwendungen. Spektrum Akademischer Verlag, Heidelberg.

[Bee00] Stadt Beelitz: Virtueller Bürgerladen, http://www.beelitz.de

[CZ4101] Computer Zeitung Nr. 41, 11. Oktober 2001.

[CZ4801] Computer Zeitung Nr. 48, 29. November 2001.

[DD98] Date, C.J.; Darwen, H. (1998): SQL - Der Standard, Addison-Wesley, Bonn.

[EIL98] Eilebrecht, Lars (1998): Apache-Web-Server, ITP, Bonn.

[EUD01] Quallcomm Website, http://www.quallcomm.com

[GSS99] Gulbins, J.; Seyfried, M.; Strack-Zimmermann, H. (1999): Dokumenten-Management. Vom Imaging zum Business-Dokument. Springer, Berlin.

[HAL01] Stadtinformationssystem der Stadt Halle(Saale), http://www.halle.de

[HY98] Hudson,T. J.; Young E.A. (1998): SSLeay: SSLeay and SSLapps FAQ. http://www2.psy.uq.edu.au/~ftp/Crypto/

[JB01] JBuilder, Borland Home Page, http://www.inprise.com

[KON01] Das Portal in die kommunale Online-Welt, http://www.kommon.de

[MEI01] Die Stadt Meißen im Internet, http://meikom.nbg.net

[MUE98] Münz, S. (1998): SELFHTML (HTML-Dateien selbst erstellen). http://www.teamone.de/selfaktuell

[MWG+99] Muth, P.; Weissenfels, J.; Gillmann, M.; Weikum, G. (1999): Integrating Light-Weight Workflow Management Systems within Existing Business Environments. In Proceedings of the 15th International Conference on Data Engineering, March 1999, Sydney, Australia.

[NC01] The Netcraft Web Server Survey, http://www.netcraft.com

[NSL01] Netcsape: Secure Socket Layer Spezifikation. http://www.netscape.com/eng/

ssl3/

[PGP01] Pretty Good Privacy, http://www.pgpi.org

[PHP01] PHP, http://www.php.net

[PSQL01] PostreSQL, http://www.postgresql.org

[Reg01] Regulierungsbehörde für Telekommunikation und Post; Digitale Signatur, http://www.regtp.de

[SeSp01] Security Space (2001): Web Server Survey. http://www.securityspace.com

[SG00] Seemann, J.; Gudenberg, J.W.v. (2000): Software-Entwurf mit UML. Springer, Berlin, Heidelberg.

[Smi98] Smith, R.E. (1998): Internet-Kryptographie. Addison-Wesley, Bonn.

[Ste99] Steinkopf, U. (1999): Aufbau eines kommunalen Onlinedienstes -Bürgerseite-. Diplomarbeit, HTWK Leipzig.

[Sus01] SuSE, http://www.suse.de

[Tan98] Tanenbaum, A.S.(1998): Computernetzwerke. Prentice Hall, München.

[TCX01] TCX, MySQL Homepage, http://www.tcx.se

[WfMC99] Workflow Management Coalition (1999): Basic Terminology & Glossary. Workflow Management Coalition, http://www.wfmc.org

[Wie99] Wiebigke, S. (1999): Aufbau eines kommunalen Onlinedienstes -Verwaltungsseite-. Diplomarbeit, HTWK Leipzig.

Abbildungsverzeichnis

Tabellenverzeichnis

Anhang

Der Anhang enthält ausgewählte Skripte zur Demonstration der Implementierung. Die beiliegende CD enthält alle Skripte.

```
<?
/*********************************************************/
/* inc/db.inc                                            */
/* Include zum Aufbau der Datenbankverbindung            */
/* Hier werden die Verbindungsdaten für alle             */
/* Datenbankzugriffe der Bürger angegeben                */
/* Autor: Daniel Heinze                                  */
/* Datum: 01.12.2001                                     */
/*********************************************************/

  $database="test";// zu benutzende Datenbankinstanz
  $sqlhost="localhost";// Rechner auf dem die Datenbank läuft
  $sqluser="root";// Login
  $sqlpass="";

  $cur=mysql_connect($sqlhost,$sqluser,$sqlpass) OR DIE ("Couldn't connect to MySQL server!");
  mysql_select_db($database) OR DIE("Couldn't select database!");
?>
```

```
<?
session_start();
session_register("user");
/*********************************************************/
/* inc/buerger_session.inc                               */
/* Include                                               */
/* regelt Authentifikation Bürger                        */
/* Autor: Daniel Heinze                                  */
/* Datum: 01.12.2001                                     */
/*********************************************************/

include "../inc/db.inc";

if(($user=="") and ($fLogin)){
    // Login und Passwort wurden vom Bürger eingegeben und müssen nun überprüft werden...

  $sql="SELECT passwort,name,vorname,strasse,plz,ort,telefon,fax,email FROM buerger WHERE login='$fLogin' AND
status='a'";

  $anfrage = mysql_query($sql,$cur);
  if (mysql_num_rows($anfrage)==1){


    $result = mysql_fetch_row($anfrage);

    if ($fPasswort==$result[0]){


        $ASTel = $result[6];
        $ASName= $result[2].' '.$result[1];
        $ASAdr= $result[3].', '.$result[4].' '.$result[5];
        $ASFax= $result[7];
        $ASMail= $result[8];
        $Nutzer=$fLogin;
        $user=$fLogin;
        $error="";
    }else{
    $error = "Falsches Passwort!";
    }

  }else{
     $error = "Sie sind noch kein aktivierter Nutzer des Systems!\n
                 Sollten Sie schon einen Nutzerantrag gestellt haben,
                 bitten wir um etwas Geduld ihre Daten m&uuml;ssen erst
                 kontrolliert werden. Ansonsten klicken Sie \n
                 <a href='neuer_buerger.php?action=neu'>hier</a>.";
  }


}
if (($user=="") and ($fLogin=="")) or ($error!="")) {
    // beim Ersten Aufruf sind keine Variablen belegt. Zuerst werden Login und Passwort abgefragt...
  echo ("<html><body>");
  echo ("<form method='post'>");
  echo $error;
  echo ("<table border=0>");
  echo ("<tr><td>Login: </td><td><input type='text' name='fLogin' size='15'><br></td></tr>");
  echo ("<tr><td>Passwort: </td><td><input type='password' name='fPassword' size='15'><br></td></tr>");
  echo ("<tr><td><input type='reset' value='abbrechen'></td><td><input type='submit' value='login'><br></
td></tr>");
  echo ("</table>");
  echo ("</form>");
```

```
  echo ("</body></html>");
  $error="Sie sind nicht eingeloggt.";
}
?>
<?php
/*********************************************************/
/* bin/neuer_antrag.php                                  */
/* Administration                                        */
/* zeigt Antragsformular zum ausfüllen für den Bürger an */
/* Autor: Daniel Heinze                                  */
/* Datum: 01.12.2001                                     */
/*********************************************************/
include "../inc/buerger_session.inc";

if($error==""){

   include "$url";
   include "../inc/buerger_action.inc";
}else{
   echo "<p>".$error."<br>";
}
?>
</body>
</html>

-------------------------------------------------------------------------------------------------------------------

<html>
<body>
<?php
/*********************************************************/
/* bin/dynascript2.php                                   */
/* Verwaltung/ Bearbeitung der Anträge                   */
/* Die vom Bürger eingegebenen Daten werden von diesem   */
/* Skript ausgewertet und in die zugehörige Antragstabelle */
/* gespeichert.                                          */
/* erwartete Parameter:tabelle                           */
/* Autor: Daniel Heinze                                  */
/* Datum: 01.12.2001                                     */
/*********************************************************/

// Es wird über: <form action="dynascript2.php" method=post> // im Formular aufgerufen.
// Dabei ist es notwendig, dass ein Eingabefeld des Formulars den Namen
// "tabelle" trägt.
// Der Inhalt dieses Feldes, also der Name der anzulegenden Tabelle sollte an
// den Antragsnamen
// angelehnt sein.
// Der Name des Antrages kann ebenfalls in einem Feld übergeben werden dieses
// muß als Name "antragsname" haben
//
// Am Ende werden die Formulardaten in die dafür vorgesehene Tabelle
// gespeichert.
//

  if ($HTTP_POST_VARS)
      {
      for(reset($HTTP_POST_VARS); $key = key($HTTP_POST_VARS); next($HTTP_POST_VARS)){
          if (is_array($HTTP_POST_VARS[$key])){
              $num=count($HTTP_POST_VARS[$key]);
          }else{
                  //print  "$$key = ".$HTTP_POST_VARS[$key]. "<br>\n";
                  if (($key != "tabelle") AND ($key != "antragsname") and ($key != "ASName") AND ($key !=
"ASAdr")){
                    // $columns wird für create table benutzt
                    $columns.="$key tinytext, ";
                    // $columns2 wird zum einfügen in die Tabelle benutzt
                    $columns2.="$key, ";
                    $values.="'$HTTP_POST_VARS[$key]', ";
                  }
              }
      }

  }
  $columns=substr($columns,0,-2);
  $columns2=substr($columns2,0,-2);
  $values=substr($values,0,-2);

  if ($tabelle){
// ------------- Datenbankteil -----------------
    $database="test";
    $sqlhost="localhost";
    $sqluser="root";
    $sqlpass="";

    $cur=mysql_connect($sqlhost,$sqluser,$sqlpass) OR DIE( "Couldn't connect to MySQL server!");
    mysql_select_db($database) OR DIE( "Couldn't select database!");
    $cur2=mysql_connect($sqlhost,$sqluser,$sqlpass) OR DIE( "Couldn't connect to MySQL server!");
    mysql_select_db($database) OR DIE( "Couldn't select database!");

// ------------- Tabellentest -----------------
    $query  = "desc $tabelle";
    if (($inhalt = mysql_query($query,$cur))==2) {
       // print "Tabelle vorhanden!<br>";

    // Jetzt werden die Formulardaten in die Tabelle gespeichert
   if (!$fehler) {

       // Finde ndx der Antragsart
       $var_sql="select ndx from antragstabellen where tabellenname = '$tabelle'";
       $inhalt = mysql_query($var_sql,$cur);
       if ($result=mysql_fetch_row($inhalt)){
           $Antrag_art=$result[0];
           echo $Antrag_ndx;
       }else{
           $fehler="Fehler beim Datenbankzugriff (kein ndx from Tabellennamen)!";
       }
   }

   if (!$fehler) {
       $adate=date("Y-m-d H:i:s");
```

```
       $var_sql="Lock Tables $tabelle WRITE, eingang WRITE";
       if (!($inhalt = mysql_query($var_sql,$cur))){
          echo "Fehler bei Lock Tables!";
       }
       $var_sql="insert into $tabelle ($columns2) VALUES ($values)";

       if (!($inhalt = mysql_query($var_sql,$cur))){

            echo("Es ist ein Fehler beim Speichern der Antragsdaten aufgetreten!");

            echo("\n<!-- $var_sql -->");
          }else
              $var_sql="select max(auftragnr) from $tabelle";
              if ($inhalt = mysql_query($var_sql,$cur)){
                  if ($result=mysql_fetch_row($inhalt)){
                      $A_ndx=$result[0];
                  }
              }else{
                  $fehler="Fehler beim Datenbankzugriff (max(ndx))!";
              }
              $adate=date("Y-m-d H:i:s");
              $var_sql="insert into eingang (date,antrag_art, antrag_nr) VALUES
                            ('$adate','$Antrag_art','$A_ndx');";
              if ($inhalt = mysql_query($var_sql,$cur)){
                  $var_sql="select max(ndx) from eingang";
                  if ($inhalt = mysql_query($var_sql,$cur)){
                      $result=mysql_fetch_row($inhalt);
                      $eingangsndx=$result[0];
                  }
                  // echo("Daten gespeichert.");
              }else{
                  $fehler="Fehler beim Speichern im Eingang!";
              }
       }
  }
  echo $fehler;

    $var_sql="Unlock Tables";
    $inhalt = mysql_query($var_sql,$cur);
}

/* Mail an Rolle */

// Suche zum Antrag zugeordnete Rolle und benachrichtige Mitarbeiter
 $var_sql="select a.antragsname, r.email from antragstabellen a left join rollen r on a.zustaendig=r.ndx where
tabellenname='$tabelle'";
 $inhalt = mysql_query($var_sql,$cur);
 if ($inhalt = mysql_query($var_sql,$cur)){
                 $result=mysql_fetch_row($inhalt);
                 $empfaenger=$result[1];
                 $antragsname=$result[0];
 }
 $mailbody="Neue Daten für\n".$antragsname."\nKlicken Sie <a href='http://127.0.0.1/bin/
verwaltung_anzeige.php'>hier</a>";

 @mail($empfaenger, "Es liegen neue Antragsdaten vor", $mailbody);

 echo "<h2>Best&auml;tigung</h2>\n
 Ihre Antragsdaten wurden gespeichert.\n<p>
 F&uuml;r R&uuml;ckfragen k&ouml;nnen Sie eine E-Mail an $empfaenger schicken.\n";

// echo "Mail an '$empfaenger' wegen $antragsname. <a href='verwaltung_anzeige.php?tabelle=$tabelle&antra-
gnr=$A_ndx&eingang_nr=$eingangsndx'>klick</a>";
?>
</body>
</html>

-----------------------------------------------------------------------------------------------------------------

<?php
session_start();
?>
<html>
<body>
<?
/*************************************************************/
/* bin/verwaltung_anzeige.php     */
/* Verwaltung/ Bearbeitung der Anträge                       */
/* Anzeige des gewünschten Datensatzes mit der Möglichkeit */
/* der Darstellung im zugehörigen Antrag.inc     */
/* erwartete Parameter:tabelle,antragnr, eingang_nr          */
/* Autor: Daniel Heinze                                      */
/* Datum: 01.12.2001                                         */
/*************************************************************/

  include "../inc/verwaltung_session.inc";

if (!$error){

    // ************************
    // Überprüfe Parameter eingang_nr

    if ($eingang_nr==""){
        echo "Suche Antrag ... ";
        $var_sql="select e.ndx
                        from eingang e left join antragstabellen a on
                        e.antrag_art=a.ndx
                        where antrag_nr='$antragnr' and a.tabellenname='$tabelle'";
        $inhalt = mysql_query($var_sql,$cur);
        if ($result=mysql_fetch_row($inhalt)){
          $eingang_nr=$result[0];
          echo "OK. Eingangsnummer $eingang_nr.";
        }else{
          echo "Dieser Antrag konnte nicht gefunden werden!\n<!-- $var_sql -->";
        }

    }

    // ************************
    // prüfe ob schon ein anderer Mitarbeiter diesen Antrag bearbeitet hat.
```

```
    $var_sql="select mitarbeiter, datum from bearbeitung where eingangsnummer='$eingang_nr'";
    $inhalt = mysql_query($var_sql,$cur);
    if ($result=mysql_fetch_row($inhalt)){
      $bearbeiter=$result[0];
      $bearbeitungszeit=$result[1];
      echo "<h3>Diesen Antrag hat bereits $bearbeiter am $bearbeitungszeit bearbeitet!<h3>";
    }

    // ************************
    // zeige Daten unformatiert an

  echo("<table border=0 cellspacing=10>\n");
         $sql="select * from $tabelle where auftragnr='$antragnr'";
         $inhalt = mysql_query($sql,$cur);
         echo ("<Form action='antrag_anzeige.php' method='post'>\n");
         echo("<table border=0><tr>\n");
         $i=0;
         $spaltenanz=mysql_num_fields($inhalt);
         if ($result=mysql_fetch_row($inhalt)){
            while ($i < $spaltenanz){

                echo ("<tr><td>".mysql_field_name($inhalt,$i)."</td><td>".$result[$i]."</td></tr>\n");
                echo ("<tr><td><input type='hidden' name='".mysql_field_name($inhalt,$i)."' value='".$re-
sult[$i]."'></td></tr>\n");
                if (strtoupper(mysql_field_name($inhalt,$i))=="LOGIN"){
                $login=$result[$i];
                }
                $i++;
            }


          }else{
                $error = "Es ist ein Fehler bei der Datenbankanfrage aufgetreten!\n<!-- $sql -->";
          }

          echo "</table>\n";
          if (!$error){
             print("<input type='hidden' name='tabelle' value='$tabelle'>\n
             <input type='hidden' name='eingang_nr' value='$eingang_nr'>\n
             <input type='hidden' name='login' value='$login'>\n
             <input type='submit' value='ansehen'>\n");
          }
          echo "</form>\n";
          echo $error;
}
?>

</body>
</html>
```

```
# ********************************************************
#  sql/createDbObjects.sql                               *
#  Datenbankobjekte                                      *
#                                                        *
#  Autor: Daniel Heinze                                        *
#  Datum: 01.12.2001                                     *
# ********************************************************
#
# Table structure for table 'antragstabellen'

CREATE TABLE antragstabellen (
   ndx mediumint(8) unsigned NOT NULL auto_increment,
   tabellenname varchar(50) NOT NULL,
   antragsname varchar(255) NOT NULL,
   zustaendig varchar(20),
   url tinytext,
   PRIMARY KEY (ndx),
   UNIQUE antragsname (antragsname)
);

# ********************************************************
#
# Table structure for table 'bearbeitung'

CREATE TABLE bearbeitung (
   ndx mediumint(8) unsigned NOT NULL auto_increment,
   mitarbeiter varchar(20) DEFAULT '0' NOT NULL,
   eingangsnummer mediumint(8) unsigned DEFAULT '0' NOT NULL,
   bemerkung varchar(255),
   abgeschlossen char(1) DEFAULT 'n' NOT NULL,
   datum datetime DEFAULT '0000-00-00 00:00:00' NOT NULL,
   PRIMARY KEY (ndx)
);

# ********************************************************
#
# Table structure for table 'buerger'

CREATE TABLE buerger (
   login varchar(15) NOT NULL,
   passwort varchar(8),
   name tinytext NOT NULL,
   vorname tinytext NOT NULL,
   strasse tinytext NOT NULL,
   plz varchar(5) NOT NULL,
   ort tinytext NOT NULL,
   telefon varchar(15),
   fax varchar(15),
   email tinytext,
   status char(1) DEFAULT 'n' NOT NULL,
   PRIMARY KEY (login)
);
```

```
# --------------------------------------------------------
#
# Table structure for table 'eingang'

CREATE TABLE eingang (
   ndx mediumint(8) unsigned NOT NULL auto_increment,
   antrag_nr mediumint(8) unsigned DEFAULT '0' NOT NULL,
   antrag_art mediumint(8) unsigned DEFAULT '0' NOT NULL,
   date datetime DEFAULT '0000-00-00 00:00:00' NOT NULL,
   PRIMARY KEY (ndx)
);

# --------------------------------------------------------
#
# Table structure for table 'mitarbeiter'

CREATE TABLE mitarbeiter (
   login varchar(20) NOT NULL,
   passwort varchar(10) NOT NULL,
   anrede varchar(4),
   name varchar(35) NOT NULL,
   vorname varchar(35),
   email varchar(60),
   telefon varchar(20),
   rolle smallint(5) unsigned,
   PRIMARY KEY (login)
);

# --------------------------------------------------------
#
# Table structure for table 'rollen'

CREATE TABLE rollen (
   ndx mediumint(8) unsigned NOT NULL auto_increment,
   r_name varchar(20) NOT NULL,
   email varchar(60),
   telefon varchar(20),
   PRIMARY KEY (ndx)
);

# --------------------------------------------------------
#
# Table structure for table 'lagerfeuer'

CREATE TABLE lagerfeuer (
   auftragnr int(11) unsigned NOT NULL auto_increment,
   login tinytext NOT NULL,
   ddat varchar(10),
   ort1 tinytext NOT NULL,
   ort2 tinytext,
   grund tinytext NOT NULL,
   beg tinytext NOT NULL,
   fin tinytext NOT NULL,
   PRIMARY KEY (auftragnr)
);
```

Erklärung

Ich versichere, dass ich die vorliegende Arbeit selbständig und nur unter Verwendung der angegebenen Quellen und Hilfsmittel angefertigt habe.

Leipzig, Februar 2002 _____

Unterschrift