Entwurf und Implementierung eines Internet-basierten *Material-Informationssystems*

Oliver Radfelder(A), Martin Gogolla(A) und Volker Behr(B)

(A)Universität Bremen, FB 3, Informatik
Postfach 330440, D-28334 Bremen

(B)Verbund Werkstoff Labor Bremen
Fahrenheitstr. 7, D-28359 Bremen

Zusammenfassung

Das Projekt *Material Informationssystem* hatte das Ziel, ein System zu entwickeln, mit dem Personen und Institutionen aus dem Bereich Verbundwerkstoffe auf einen gemeinsamen Produktdatenbestand zugreifen können. Wir haben ein System entwickelt, das den Zugriff über das Internet ermöglicht. Als Benutzungsschnittstelle wird lediglich ein Webbrowser vorausgesetzt anstatt eine vollständige Applikation auf den Rechnern der Teilnehmer zu installieren. Während der ersten Projektphase wurden die Anforderungen an ein solches neues System analysiert und ein Prototyp entworfen und implementiert. In der zweiten Projektphase wurde das System als Server mit Hilfe von SQL und Java implementiert.

Inhaltsverzeichnis

1 Einleitung

Das Projekt *Material-Informationssystem* hatte das Ziel, in Kooperation mit dem Verbund Werkstoff Labor Bremen ein System zu entwickeln, mit dem externe Teilnehmer sowohl auf den Datenbestand anderer Teilnehmer Zugriff nehmen können, als auch ihren eigenen Datenbestand selbständig pflegen können. Der letztere Aspekt wird in naher Zukunft von zunehmender Bedeutung sein: Strukturierte Informationen ähneln sich branchenspezifisch so sehr, daß es unsinnig und vor allem zu teuer erscheint, für jeden einzelnen Anbieter eigenständige Informationssysteme zu entwickeln. Bis heute ist es daher gängige Praxis, Softwarepakete als Gesamtlösung anzubieten, die an die jeweilige Umgebung angepaßt werden. Die Probleme, die ein solches Vorgehen mit sich bringt, liegen auf der Hand: Jedes einzelne System muß eigenständig verwaltet werden. Die daraus resultierenden Kosten übersteigen zunehmend die Anschaffungskosten, da teure Spezialisten für Datenbanksysteme, Rechnernetze und Design beschäftigt werden müssen.

Eine weitere Alternative ist, Informationssysteme zu zentralisieren und die Aufnahme und Pflege der Daten ebenfalls an zentraler Stelle vorzunehmen. So sind Internetauftritten von Städten oder Gemeinden [bre] darauf angewiesen, Informationen von Geschäften und lokalen Organisationen selbständig zu sammeln und einzugeben. Ein solches Vorgehen würde eine einzelne Organisation personell überlasten, sobald die abrufbaren Informationen komplexer strukturiert sein sollen. In unserem Projekt sollen beispielsweise je nach Kategorie unterschiedliche Kriterien für einzelne Produkte anzugeben sein.

Es wäre von großem Vorteil, wenn auf der einen Seite die technische Verwaltung eines Informationssystems, an dem mehrere Anbieter beteiligt sind, zentral administriert wird, auf der anderen Seite die einzelnen Anbieter aber die Pflege ihrer Daten eigenverantwortlich übernehmen. Um eine solche Umgebung bereitstellen zu können, muß es möglich sein, daß einzelne Anbieter ohne technischen Aufwand - wie er beispielsweise durch die Installation von Software entsteht - ihre Angebotsdaten pflegen. Der zur Zeit vielversprechendste Ansatz ist, das Internet als Kommunikationsmedium und Webbrowser als Schnittstelle zur Datenpflege einzusetzen.

Wenn allerdings mehrere Anbieter ihr Informationsangebot in einem System verwalten, erfordert die Verwaltung der Daten und deren Konsistenzhaltung einen so hohen Aufwand, daß die Verwendung eines Datenbanksystemes gerechtfertigt ist.

Komplexe Strukturen, wie sie in branchenspezifischen Anwendungsbereichen zwangsläufig entstehen, sind mit dem objektorientierten Ansatz leichter nachzubilden als mit relationalen Strukturen, zumal modernere Programmiersprachen wie Java oder C++ selbst objektorientiert sind. Objektorientierte Datenbanksysteme sind jedoch noch nicht weit verbreitet, volle Implementierungen des Standards[CBB+97] sind selten und die wenigen Systeme, die existieren, sind teuer und haben sich noch nicht in ausreichendem Maße bewährt. Relationale Datenbanksysteme sind dagegen schon lange verfügbar und mit *SQL* [Ame92] existiert ein von nahezu allen Systemen unterstützter Standard. Nach wie vor sprechen also gute Gründe dafür, die strukturierten Daten in einer relationalen Datenbank zu speichern und die Anwendungen in einer objektorientierten Programmiersprache zu entwickeln. Der daraus resultierende Paradigmenbruch zwischen Datenbank und Anwendung wird durch das Konzept des objekt-relationalen Mappings aufgefangen.

2 Erste Projektphase: Analyse und Prototyping

Die Arbeitsgruppe Datenbanksystem des Fachbereichs Informatik der Universität Bremen hat als Mitglied des Technologiezentrums Informatik (TZI) gemeinsam mit dem Verbund Werkstoff Labor Bremen e.V. 1996 den Antrag auf *Förderung eines Projektes im Bereich der Informations- und Kommunikationstechnologien "Informationssystem/Materialdatenbank Verbundwerkstoffe"* beim Senator für Wirtschaft, Mittelstand, Technologie und Europaangelegenheiten gestellt. Der geplante Projektzeitraum von zwei Jahren sollte sich vom 1.1.1997 bis zum 31.12.1999 erstrecken. In dem Zuweisungsbescheid vom 4.3.1997 wurde den Kooperationspartnern rückwirkend für den oben genannten Zeitraum ein Betrag von 340.600 DM bewilligt. Zum 1.5.1997 wurde ein wissenschaftlicher Mitarbeiter für dieses Projekt eingestellt.

2.1 Analyse

Während der ersten Projektmonate wurde innerhalb der Analysephase zunächst die informelle Beschreibung des Anforderungsprofils konkretisiert. Dabei wurde deutlich, daß in dem Projekt eine Anforderung von zentraler Bedeutung sein würde: die Benutzungsschnittstelle sollte so einfach zu handhaben sein, daß die Benutzerin möglichst intuitiv und ohne Hinzuziehen eines Handbuchs das Programm bedienen kann. Diese Anforderung war deswegen von so zentraler Bedeutung, weil gerade im Datenbankbereich nach wie vor die Ansicht vorherrscht, Anfragen würden von Spezialisten in textueller Notation formuliert, nachdem Endanwender die Anfrage informell beschrieben haben. Grafische Anfragenotationen waren zum Zeitpunkt des Projektbeginns ebenso wie heute ein Forschungsfeld, von dem wenig Ergebnisse in kommerzielle Anwendungen eingegangen sind [CS95, Shn94, CCD+99]. Und selbst in Forschungsprototypen wird bis auf wenige Ausnahmen die visuelle Unterstützung in Anfragen sehr stark an das jeweils untersuchte Fachgebiet angelegt. Insbesondere Bibliotheken [Vee95], medizinische Anwendungen und geographische Informationssysteme [DFMR95] werden in diesen Prototypen untersucht. Bei diesem fachlich eingeschränkten Nutzerkreis wird dann jeweils davon ausgegangen, daß ebenfalls der Umgang mit Computern keine Schwierigkeiten bereitet.

In Laufe der Entwurfsphase mußte die bis zu dem Zeitpunkt als informelle Beschreibung vorliegende Basisstruktur einer Revision unterzogen und in ein höheres Datenmodell übertragen werden. Als Grundlegendes Datenmodell und Beschreibungssprache wurde zunächst das Entity-Relationship Modell [Che83, BCN92, GH91] gewählt, wie es im Datenbankentwurf üblich ist. Mit dem Erscheinen der *Unified Modelling Language* [OMG99] und deren Standardisierung durch die *Object Management Group* (OMG) im Jahr des Projektbeginns bot es sich jedoch an, sie auch in diesem Projekt als Modellierungssprache einzusetzen. Daß diese Entscheidung richtig war, zeigt sich daran, daß innerhalb kürzester Zeit die *UML* einen Verbreitungs- und Bekanntheitsgrad auch außerhalb der akademischen Welt erlangt hat, der deutlich über dem anderer Modellierungssprachen liegt. Für unser Projekt hat sich herausgestellt, daß die *UML* einen entscheidenden Vorteil hat: ihre Sprachelemente eignen sich gleichermaßen für den objektorientierten Entwurf wie für den Entwurf von relationalen Strukturen. So konnten die Schwierigkeiten, die bei dem Paradigmenbruch zwischen klassischer Datenbankmodellierung für das zugrundegelegte relationale Datenbanksystem *Oracle* und der objektorientierten Modellierung für die Anwendung entschärft werden.

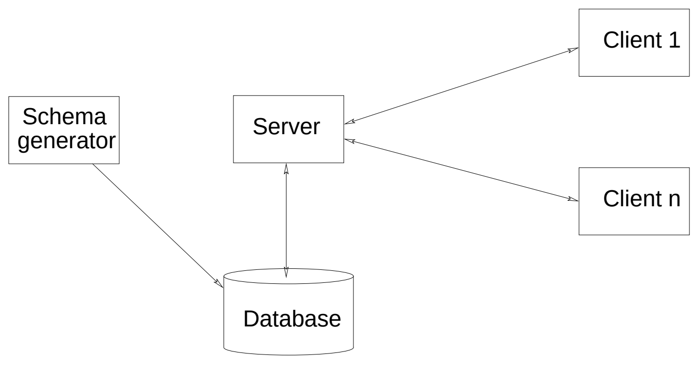

Abbildung 1: Client, Server, Datenbank und Schemagenerator

2.2 Prototyping

Bis zum dritten Quartal des Jahres 1998 wurde ein Prototyp entwickelt. Dieser Prototyp hatte zum Ziel, wesentliche Funktionalität der Oberfläche in der Endanwendung zu demonstrieren. Das Konzept für den Prototyp basierte auf einer Client/Server Architektur, so wie es zunächst auch für die endgültige Applikation vorgesehen war. Beide Teile , Client und Server, wurden als eigenständige Applikationen in der Programmiersprache Java entwickelt. Zusätzlich wurde ein Schemagenerator entworfen. Mit dem Schemagenerator wird aus einer textuellen Spezifikation ein relationales Datenbankschema und ein Mappingschema für objektorientierte Anfragen erzeugt (siehe Abb. 1). Dieses Vorgehen hat den entscheidenden Vorteil, daß das Datenbankschema nicht schon zu Beginn der Evaluierung vollständig beschrieben sein muß. Stattdessen können im Verlauf der Entwicklung neue Klassen, Assoziationen oder Attribute hinzugefügt werden. Ein Nachteil dieser Strategie ist allerdings, daß das generische Mapping von Anfragen nicht optimiert ist und damit für die endgültige Applikation nicht in Frage kommt.

Serverseitig wurde eine Applikation entworfen, mit der der Client über das TCP/IP Protokoll Kontakt aufnehmen kann, um Anfragen abzusetzen und die Ergebnisse entgegenzunehmen. Die Anfragen wurden mit Hilfe der Mappingstrategie in relationale Anfragen umgeformt und an den Datenbankserver weitergeleitet. Die Ergebnisse wurden wiederum von diesem entgegengenommen und nach Konvertierung in eine Objektstruktur über ein propriäteres Protokoll an den Client zurückgesandt.

Der Client wurde ebenfalls als eine eigenständige Applikation entworfen und bestand im wesentlichen aus einem Modul zur Formulierung von grafischen Anfragen. Wie in [RGB99] beschrieben, ist die Anfragennotation konzeptionell eng an UML angelehnt (siehe Abb. 2).

2.3 Evaluierung des Prototypen

Nachdem der Prototyp entwickelt war, hatten wir Gelegenheit, das Ergebnis in Hinblick auf Nutzerfreundlichkeit zu evaluieren. Sowohl mit Besuchern des TZI-Tags 1999 als auch mit einer ausgesuchten Benutzergruppe wurde evaluiert, inwieweit sich eine an UML angelehnte grafische Anfragesprache eignet, Gelegenheitsnutzern einen schnellen Zugriff zu geben. Dabei hat sich erge-

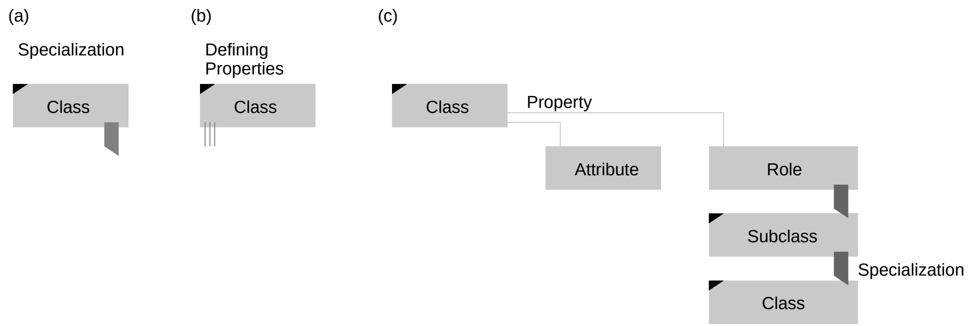

Abbildung 2: Konzepte der Anfragenotation

ben, daß in diesem Bereich noch erheblicher Forschungsbedarf existiert. Zum einen ist die UML, und damit auch unsere daran angelegte Anfragesprache, offensichtlich nicht so intuit, wie Informatiker annehmen, zum anderen ist sie Nicht-Informatikern auch noch so unbekannt, daß selbst einfache Konzepte keinen kognitiven Anknüpfungspunkt finden.

Es hat sich ergeben, daß einfache Anfragen relativ schnell zu formulieren sind. Damit hebt sich eine solche grafische Anfragesprache aber eben noch nicht in ausreichendem Maße von den üblichen textuellen Schnittstellen ab, bei denen in aller Regel lediglich nach Stichwörtern gesucht wird.

In textuellen, zwei-dimensionalen grafische Anfragesprachen wird häufig versucht, die zusätzliche Dimension zu nutzen, gedanklich nahe beieinander liegende Konzepte auch in der Konstruktion und Visualisierung von Anfragen nahe beieinander zu präsentieren. So haben wir gezeigt, daß ein Anfragegraph sowohl Selektion als auch Projektion ausdrücken kann, obwohl die Anwenderin nur einen einzelnen Graphen konstruiert. Um die Überlegenheit eines solchen Konzeptes zu erfassen, muß die Anwenderin jedoch ein tiefgreifendes Verständnis von diesen Konzepten haben, was die von uns angepeilte Zielgruppe gerade nicht haben soll. Insbesondere komplexe Selektionen, Sortierungen und strukturierte Verbünde sind nach unseren Erfahrungen zu komplex in der Bedienung.

3 Zweite Projektphase: Implementierung

Laut Änderungsbescheid vom 6.9.1999 wurde der Projektzeitraum um sechs Monate vom 1.10.1999 bis zum 31.3.2000 verlängert. In diesem Zeitraum wurde zunächst aufgrund der Erfahrungen mit dem Prototypen das Konzept überarbeitet. Es hatten sich zwei konkrete Punkte herausgestellt, die wir überarbeiten wollten:

Benutzungsführung Aufgrund der Erfahrungen in der Evaluation sollte sich die Benutzungsführung stärker an klassischen Anfragemethoden orientieren. Die innovative grafische Benutzungsschnittstelle ist offensichtlich für die Normalnutzerin noch zu wenig vertraut. Andererseits wollten wir aber auch nicht auf eine rein textuelle Schnittstelle zurückgehen, für die die Normalnutzerin üblicherweise einen Informatiker als "Übersetzer" benötigt. Als guter Kompriß hat sich eine navigationsorientierte Schnittstelle herausgestellt, die mit einer grafisch ansprechenden intuitiven Oberfläche versehen wird.

Einzelapplikationen Innerhalb der letzten Jahre hat sich die Erkenntnis durchgesetzt, daß nicht so sehr die Anschaffungskosten für Hardware und Software bei den Gesamtkosten für Informationssysteme ins Gewicht fallen. Vielmehr ist es die Pflege der Applikationen und der Aufwand, um die Anwendungen auf verschiedenen Rechnern auf dem gleichen Stand zu halten. Client-Software, die auf Kundenrechnern installiert werden muß, bevor die Kundin sie benutzen kann, ist aufwendig zu warten. Dies gilt besonders dann, wenn die Kundinnen auf unterschiedlichen Plattformen arbeiten (Macintosh, Windows, Linux etc). Daher werden moderne Architekturen so aufgebaut, daß auf Client-Seite soweit wie möglich Standardsoftware eingesetzt wird, während der Hauptteil der Applikation auf der Serverseite plaziert wird. Damit die Standardsoftware auf die applikationsspezifischen Daten zugreifen kann, wird dann serverseitig eine Schicht zwischen (Standard-) Client-Anfragen und Datenbankmanagementsystem entworfen. Diese Schicht ist der Kern der Applikation in der 3-Schichten-Architektur [Cor98].

Im Zuge der zweiten Projektphase wurden weiterhin umfangreiche Daten vom Verbund Werkstoff Labor gesammelt und in Form von Listen mit dem Computer erfaßt. Diese Datenerfassung war die Grundlage für kleinere Schemaänderungen im Laufe der Implementierung. Wir haben uns entschlossen,

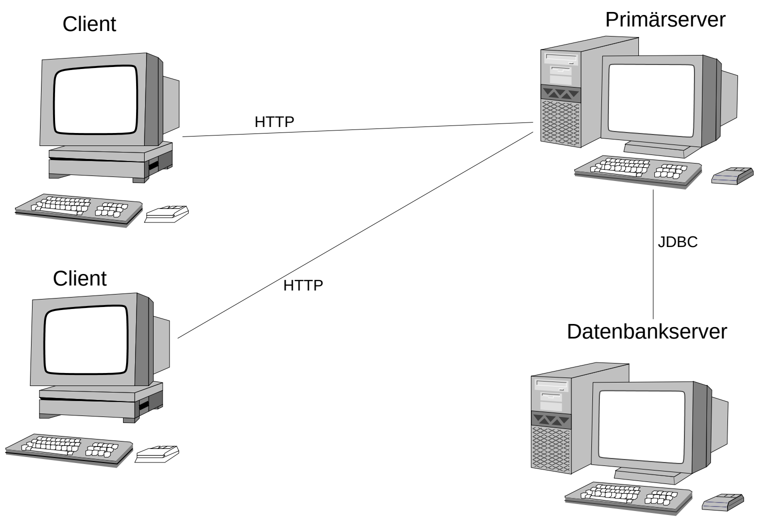

Abbildung 3: Rechnerkonstellation für eine 3-Schicht-Architektur

auf Seite des Client Standard-Webbrowser einzusetzen und die Applikation mit der Programmiersprache Java als Dienst-Server zu entwerfen. Java bietet sich hierfür aus verschiedenen Gründen an: (1) Java bietet mit dem JDBC-API [WM97] eine mittlerweile weit entwickelte standardisierte Schnittstelle zu Datenbankmanagementsystemen, mit der der Zugriff auf SQL-Datenbanken unabhängig vom konkreten System gewährleistet wird. (2) Java ist als Programmiersprache für das Internet entworfen und bietet daher eine gute Unterstützung für die Netzeinbindung.

Ein weiterer, häufig zitierter Vorteil von Java ist die Ausführbarkeit innerhalb von Webbrowsern: kleine Applikationen (sogenannte Applets), die auf dem Server installiert sind, werden auf Anfrage des Client zunächst vollständig übertragen und dann ausgeführt. So könnte die Notwendigkeit einer Zwischenschicht vermieden werden, indem die im Browser ablaufende Applikation direkt auf die Datenbank zugreift. Dieses Vorgehen hat jedoch verschiedene Nachteile: Zum einen werden nach wie vor regelmäßig Sicherheitslücken in den Implementierungen der Virtuellen Java-Maschinen der Browser entdeckt, die den übertragenen Java-Code ausführen. Dies führt zu einem steigenden Vertrauensverlust auf Seiten der Anwenderinnen, die deshalb häufig die Java-Unterstützung in ihren Browsern abschalten. Zum anderen müßte mit einer solchen Architektur die Schnittstelle des Datenbankmanagementsystems in-

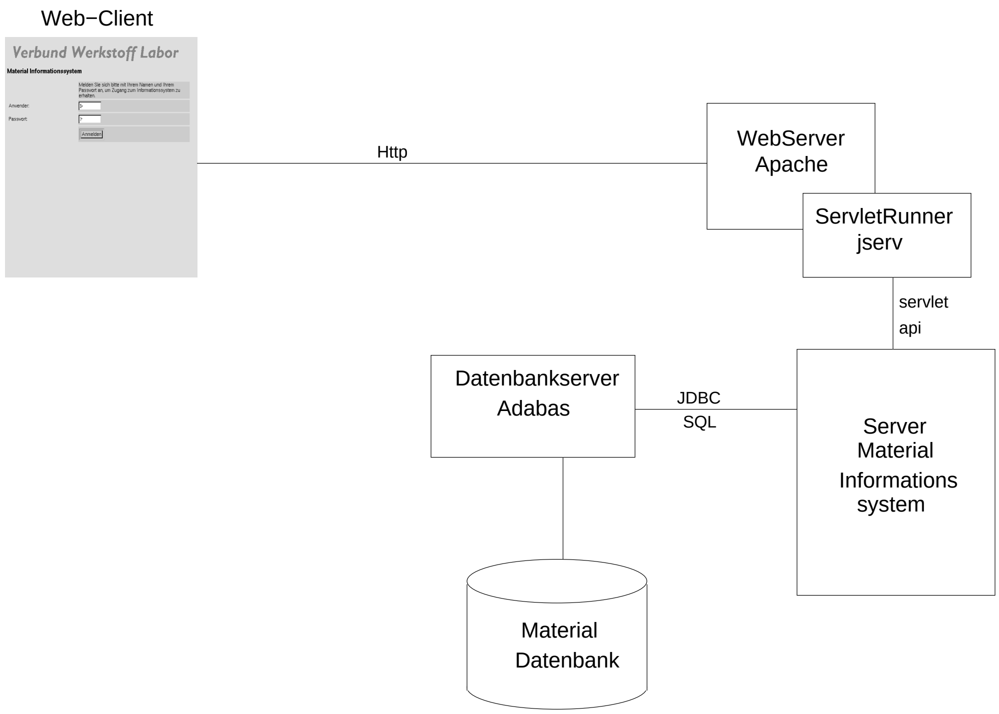

Abbildung 4: Architektur des Gesamtsystems

ternetweit offen sein, was ein hohes Maß an Sicherheitsbewußtsein auf Seiten der Betreiberin bedeutet, die regelmäßig das Datenbanksystem auf Angriffe kontrollieren müßte. Da solche Systeme zu den komplexesten der kommerziellen Software gehören, ist dies regelmäßig eine der größten Schwachpunkte auch bei großen Dienste-Anbietern im Internet (Amazon, Yahoo etc.). Solche Angriffe sind in einer 3-Schicht-Architektur deutlich einfacher abzuwehren.

Mit dem Servlet-API bietet Java eine Spezifikation, um serverseitig Java-Code auszuführen. So ist es möglich, auf der Client-Seite lediglich das sichere HTML einzusetzen, während die eigentliche Applikation, die die Anfragen entgegennimmt, Antworten generiert und Datenmanipulations-Anfragen verarbeiten, in einen Web-Server eingebettet ist.

Im Laufe der Implementierungsphase kam uns die Entscheidung zugute, auch den Server plattformunabhängig in Java zu implementieren und die Datenbankschnittstelle ebenfalls mit JDBC und Standard-SQL unabhängig von einem speziellen Server zu halten. Das ursprünglich genutze Server-System auf Basis der SPARC-Architektur von Sun entspricht heute nicht mehr dem Stand der Technik. Ein Upgrade auf ein neueres System des gleichen Herstellers ist zum einen sehr kostenintensiv, zum anderen nicht ohne umfangreiche technische Unterstützung durchzuführen. Linux war bis zu diesem Zeitpunkt aber schon so stark weiterentwickelt, daß es sich uneingeschränkt als Serverplattform für ein Projekt dieses Umfangs eignet. Ein solches PC-basiertes System ist deutlich einfacher wartbar und notfalls auch leicht durch Speichererweiterungen oder Prozessoraustausch aufzurüsten. Wir haben uns ebenfalls dazu entschlossen, das Datenbanksystem *Oracle* durch ein günstigeres zu ersetzen, da bei Oracle hohe Kosten in der Projektfolgezeit durch Wartungsverträge zu erwarten sind. Es wurde ebenfalls festgelegt, das auf Standard-SQL basierende und fast ebenso leistungsfähige System von *Adabas* einzusetzen. Als Schnittstelle zwischen dem Browser der Anwenderin und dem Material Informationssystem wird der freie Webserver *Apache* und die Servlet-Implementierung *jserv* eingesetzt (siehe Abb. 4).

Entwurf In Abbildung 5 ist der Entwurf der serverseitigen Datenhaltungskomponente als UML-Klassendiagramm gezeigt. Die Klassen *Anbieter* umfaßt alle diejenigen Teilnehmer, die *Produkte* anbieten. Jeder *Anwender* muß sich zur Benutzung des Systems autorisieren.*Produkte*, die zum Angebot eines *Anbieters* gehören, werden in eine Kategorie eingeordnet, durch die bestimmt wird welche der beschreibenden Eigenschaften für das *Produkt* angegeben werden können. Sowohl *Anbietern* als auch *Produkten* werden Dokumente zugeordnet, die vom jeweiligen *Anbieter* gestaltet werden. In dieser Form werden die Daten in der Datenbank gehalten und von dem Server - je nach Anfrage - mittels Servlets an den Web-Client ausgeliefert.

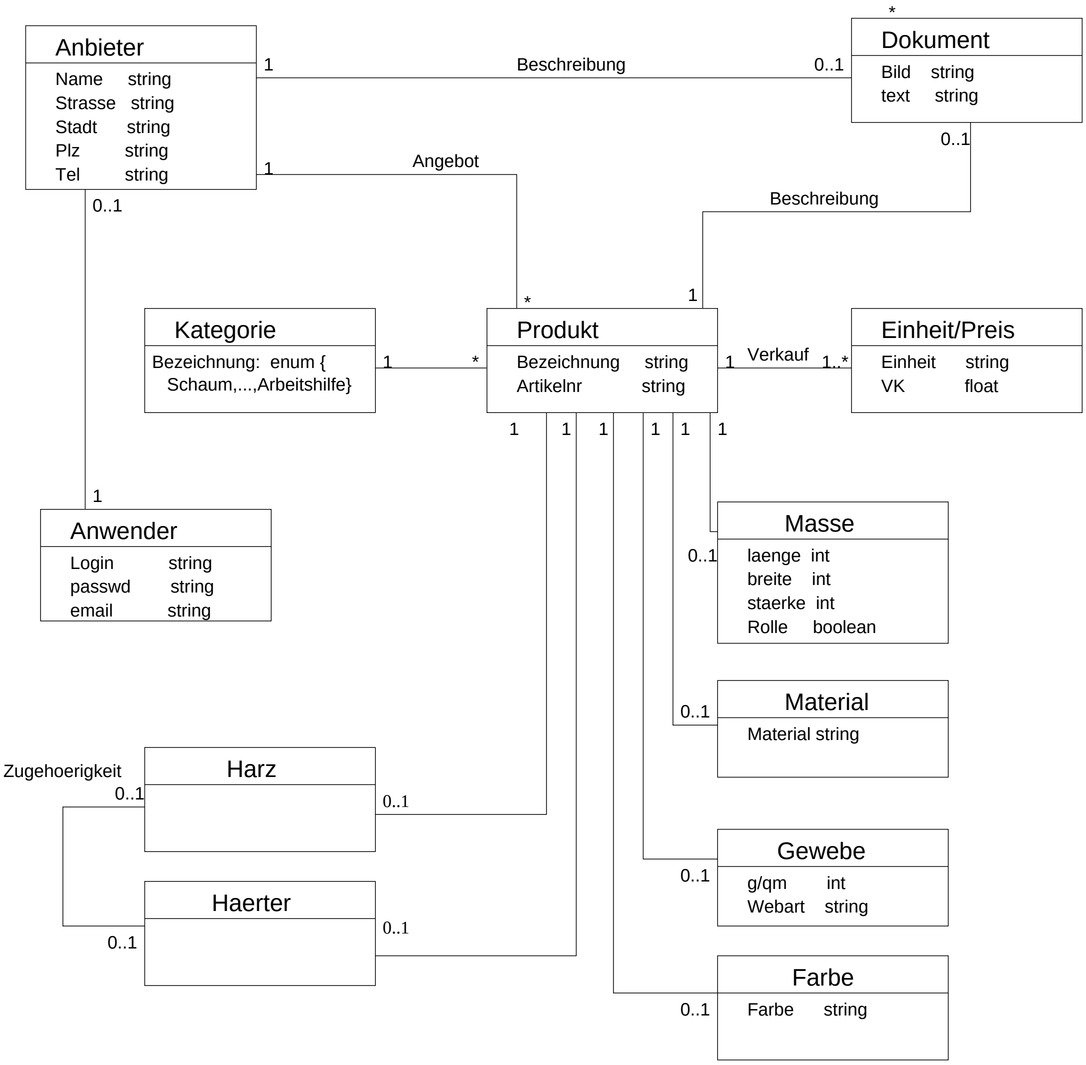

Abbildung 5: UML-Klassendiagramm

4 Benutzung

Der Zugang zum Material Informationssystem wird über einen einfachen Webbrowser wie beispielsweise Netscape gewährleistet. Zu Recht haben viele Nutzerinnen im Internet Bedenken, in ihren Browsern Java, Javascript oder Cookies anzuschalten. Dies wird von uns berücksichtigt, und der Anwender muß keine dieser Eigenschaften aktiviert haben, um das Material Informationssystem zu nutzen.

4.1 Anmeldung

Wenn die Anwenderin auf die Seite des Verbund Werkstoff Labors navigiert, wird sie von dort auf die *Anmeldungsseite* geführt.(siehe Abb. 6) Alle Teil-

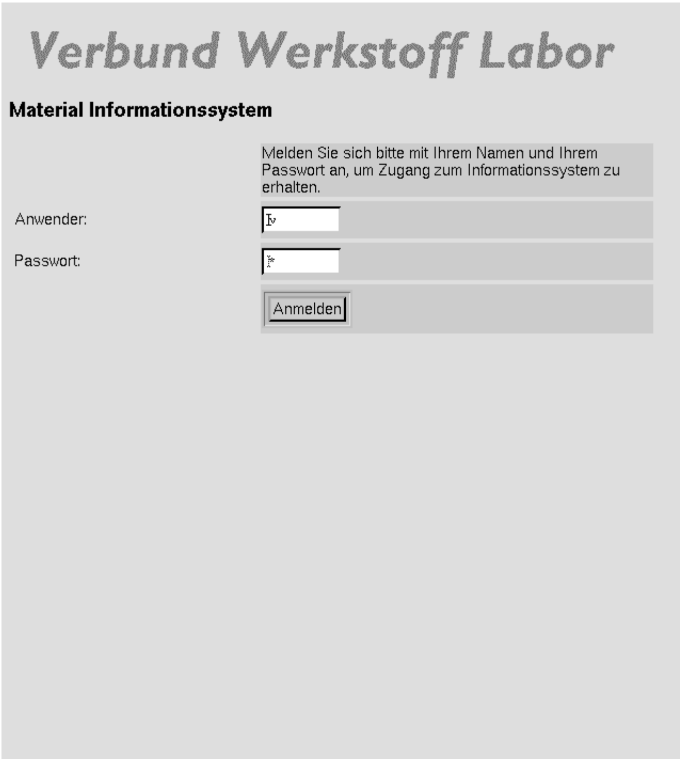

Abbildung 6: Oberfläche zur Anmeldung

nehmerinnen haben eine individuelle Kombination aus Passwort und Namen, mit der sie sich anmelden. Wenn diese Kombination auf der Anmeldungsseite eingegeben wird, wird eine *Session* erzeugt, mit der alle Interaktionen verwaltet werden, die im folgenden ausgeführt werden. Durch diesen Mechanismus wird das ansonsten zustandslose *http* Protokoll so erweitert, daß die Anwenderin nicht bei jeder Datenänderung ihre Identität belegen muß. Die folgenden Seiten sind daher auch nur zu erreichen, wenn die Anmeldung korrekt erfolgt ist.

Auf jeder Seite des Gesamtsystems ist das Corporate Design des Verbund Werkstoff Labors umgesetzt. Jede Seite hat im Kopfbereich den entsprechenden Schriftzug und die gleiche Hintergrundfarbe. Wo möglich, sind die Seiten aufgeteilt in eine linke und eine rechte Seite: links stehen Beschreibungen der rechts aufgeführten Aktionen oder Eingabefelder. Felder auf der rechten Seite sind einzeln grau unterlegt, um das Corporate Design widerzugeben. Ebenfalls auf der rechten Seite im oberen Bereich befindet sich eine kurze Beschreibung für die jeweilige Seite.

4.2 Hauptseite

Nachdem die Kombination aus Passwort und Name durch Anklicken des mit *Anmelden* beschrifteten Buttons bestätigt wurde, erscheint die Hauptseite des Systems. (siehe Abb. 7) Diese Seite ist die Zentralstelle für die wichtigsten Aktionen die die Anwenderin durchführen kann: Produkte von Anbietern abfragen, eigene Produkte aufnehmen, die eigene Präsentation gestalten und die persönlichen Daten verwalten.

4.3 Mitglied aufnehmen

Wenn die Anwenderin als Administratorin angemeldet ist, ist zusätzlich zu den oben aufgeführten Punkten ein weitere Aktion möglich: *neues Mitglied aufnehmen*. Wenn eine Anwenderin unter anderem Namen angemeldet ist, ist dieser Punkt nicht erreichbar, damit nur von dem administrierenden Verbund Werkstoff Labor neue Mitglieder aufgenommen werden können.

In Abbildung 8 ist die Eingabemaske zu sehen, in der ein neues Mitglied aufgenommen wird. Diese Stammdaten beinhalten die Zugangsdaten *Login* und *Passwort*, die dem neuen Mitglied mitgeteilt werden, später aber von die-

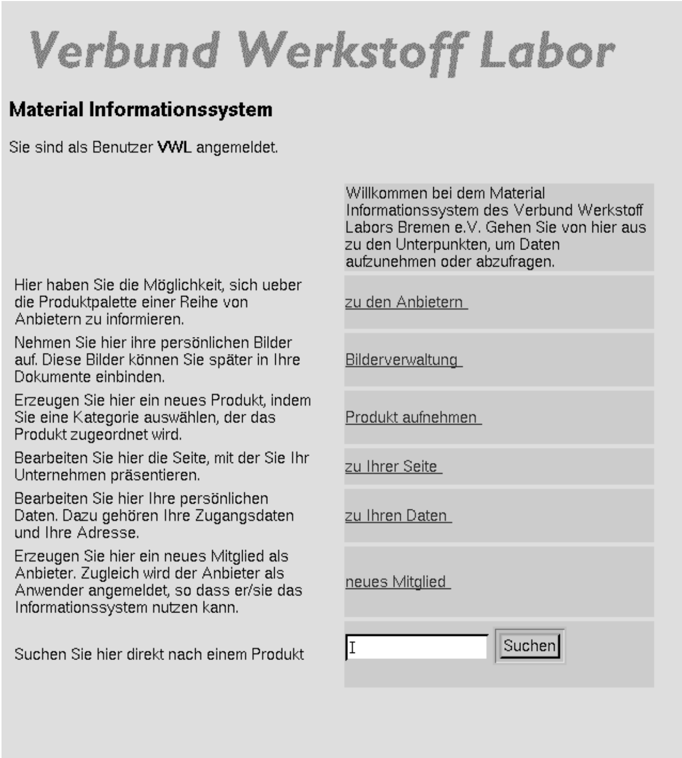

Abbildung 7: Hauptseite des Informationssystem

sem beliebig geändert werden können. Die restlichen Daten, im wesentlichen Adreßdaten - werden bei automatisch generierten Produktpräsentationen und in Anfrageergebnissen benötigt.

Eine ähnliche Seite - mit dem gleichen Layout, aber anderem Erklärungstext - erreicht die Anwenderin von der Hauptseite, wenn sie dem Verweis *zu Ihren Daten* folgt.

Geänderte oder eingefügte Daten werden durch Anklicken des *Bestätigen* Aktionsfeldes in die Datenbank übernommen.

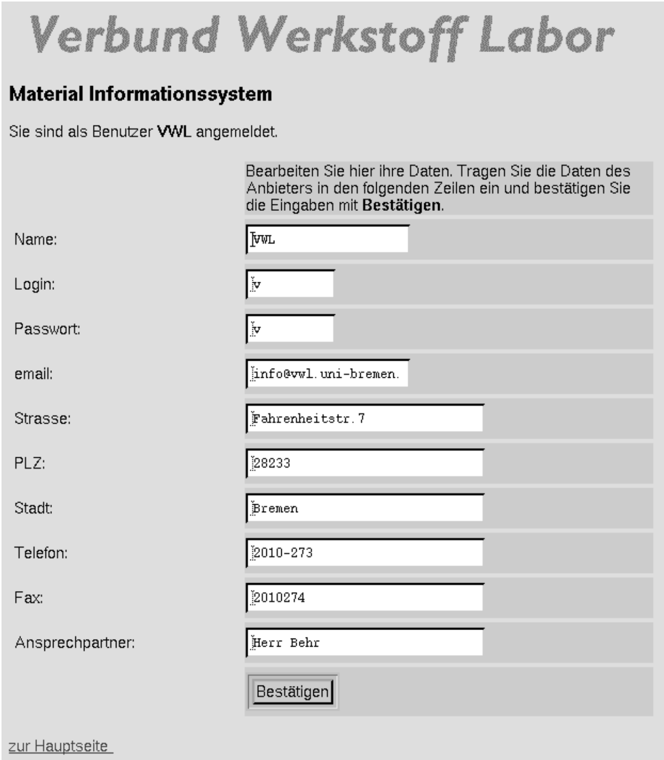

Abbildung 8: Eintragen eines Mitglieds und Bearbeiten der Daten

4.4 Produkte abfragen

Um auf Produkte zuzugreifen, wird in erster Linie ein *navigierender Ansatz* verfolgt. *Anbieter* bieten *Produkte* in verschiedenen *Kategorien* an. *Produkte* bestehen aus Eigenschaften und haben Beziehungen zu *Einheiten* mit *Preisen*.

Diese Art der mengenorientierten Objektrefenzierung wird vom objektorientieren Ansatz in hohem Maße unterstützt. Die Umsetzung in eine Webgestützte Oberfläche wird durch generierte HTML-Seiten mit Verweisen auf die jeweiligen Komponenten vorgenommen. Der große Vorteil dieses Herangehens besteht darin, daß ein Anwender keine komplexen textuellen Anfragesprachen wie *SQL* beherrschen muß, sondern den intuitiven Verbindungen zwischen Objekten folgt.

Abbildung 9: Eine exemplarische Liste der am System beteiligten Anbieter

4.4.1 Auswahl einer Anbieterin

Führt die Anwenderin die Aktion *zu den Anbietern* auf der Hauptseite aus, gelangt sie zu einer Seite mit den Anbieterinnen. Diese Seite wird aus dem Resultat einer Datenbankabfrage generiert, muß also nicht bearbeitet werden, um sie auf aktuellem Stand zu halten. In der tabellarischen Darstellung werden einige Stammdaten des jeweiligen Anbieters aufgeführt, um der Anwenderin die Möglichkeit zu geben, Kurzinformationen zu der jeweiligen Anbieterin einzusehen. Dazu gehören beispielsweise *Name*, *Strasse* und *Ansprechpartner*.

Von jedem einzelnen Eintrag sind zwei Aktionen möglich. Zum einen kommt die Anwenderin direkt zur Präsentationsseite des Anbieters, wenn der entsprechende Name in der ersten Spalte angewählt wird. Dort präsentiert sich

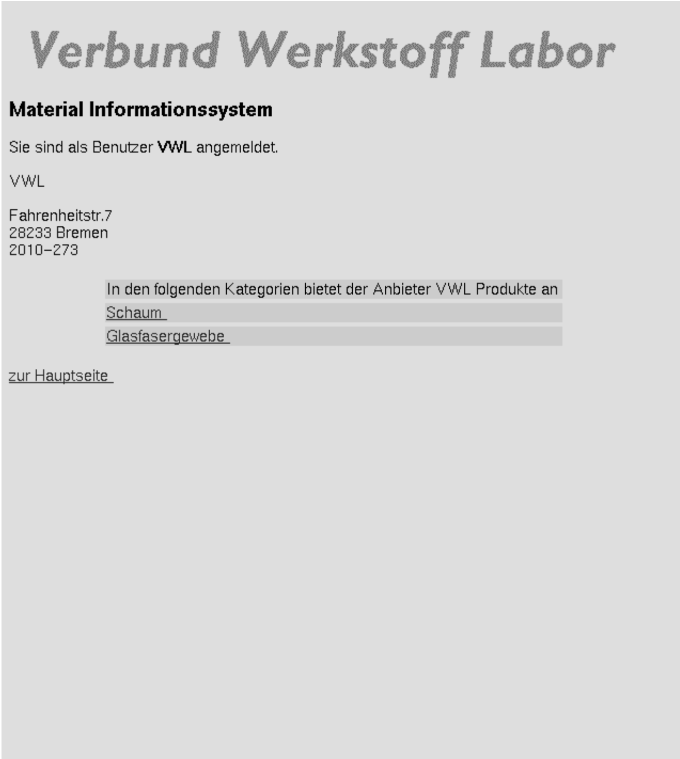

Abbildung 10: Die Liste der Kategorien, in denen ein Anbieter Produkte anbietet

das Unternehmen mit einem Dokument, das ebenfalls aus der Datenbank generiert wird.

Wenn die Anwenderin dagegen den Hyperlink *Angebot* eines speziellen Anbieters wählt, wird auf der folgenden Seite eine Kategorie ausgewählt.

4.4.2 Auswahl einer Kategorie

Auf dieser Seite wählt die Anwenderin, aus welcher Kategorie Produkte betrachtet werden sollen. Im oberen Bereich werden einige Stammdaten des gewählten Anbieters angezeigt. Darunter wird eine generierte Liste derjenigen Kategorien dargestellt, in denen der Anbieter mindestens ein Produkt anbietet. Wenn jetzt eine Kategorie durch Anklicken des Namens ausgewählt

Verbund Werkstoff Labor

Material Informationssystem

Sie sind als Benutzer **VWL** angemeldet.

Produkte der Kategorie Schaum des Anbieters:

VWL

Fahrenheitstr.7
28233 Bremen
2010–273

Bezeichnung	Artikelnr	g/qm	Laenge	Breite	Staerke	Einheit	Ein./Preis	Preis
Schaum3	10255	10	2000	1000	100	10 Rollen	1 Rolle	20,00
						20 Rollen	1 Rolle	18,33
Schaum 2	10356	10	2000	1000	200	1 Rolle	1 Rolle	10,00

zur Hauptseite

Abbildung 11: Die Liste der Produkte einer Kategorie

wird, wird eine Seite aufgebaut, die in tabellarischer Darstellung alle Produkte auflistet, die der gewählte Anbieter im Angebot hat.

4.4.3 Produktliste

Auf dieser Seite werden alle Produkte in tabellarischer Form aufgelistet, die in der aktuellen Kategorie angeboten werden. Je nach Kategorie werden verschiedene Eigenschaften angegeben. So hat ein *Schaum* beispielsweise die Angabe *g/qm*, während Produkte der Kategorie *Glasfasergewebe* Angaben über die Webart enthalten können.

Allen Produkten ist gemeinsam, daß Preisinformationen je nach Menge variieren können. In der tabellarischen Form drückt sich das dadurch aus, daß die letzten drei Spalten jeweils eine Menge/Preiskombination beinhalten. Ist mehr als eine Kombination enthalten, werden sie zeilenweise unter der jeweiligen Produktspalte aufgeführt. (siehe Abb. 10)

Ein einzelnes Produkt kann hier ausgewählt werden, worauf die entsprechende Produktseite angezeigt wird. Eine solche Produktseite zeigt immer die

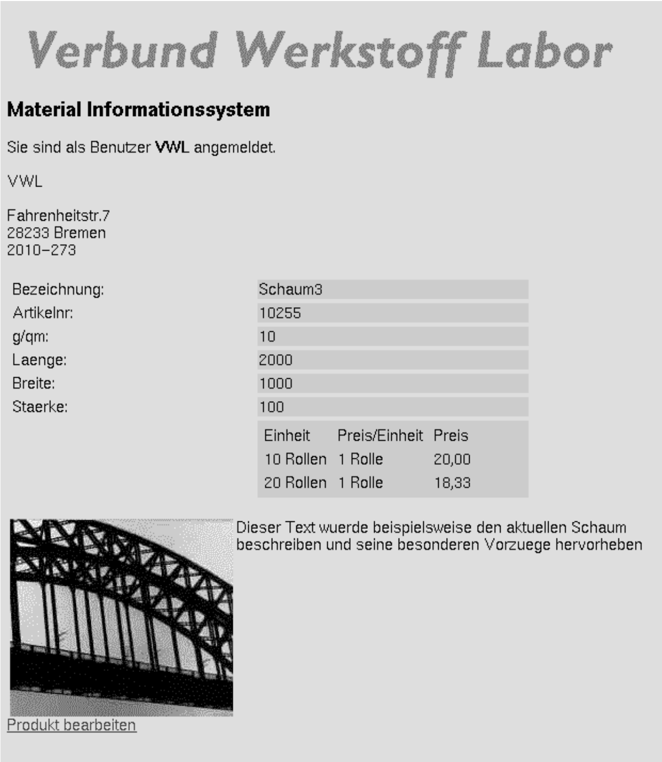

Abbildung 12: Die Produktseite eines einzelnen Produktes

gesamten Daten, die zu einem Produkt verfügbar sind. Zusätzlich können mehrere Absätze mit Grafiken zur Beschreibung verfügbar sein.

4.5 Datenbearbeitung

Wenn eine Anbieterin in dem System aufgenommen ist (siehe Abschnitt 4.3), kann sie sich mit dem ihr zugewiesenen Namen und Paßwort anmelden. Ihre

Stammdaten kann sie beliebig ändern, indem von der Hauptseite die Aktion *zu Ihren Daten* ausgeführt wird. Die darauf erscheinende Seite ermöglicht, die einzelnen Felder wie in Abbildung 8 zu bearbeiten. Durch Anklicken der Aktionsfläche *Bestätigen* werden die veränderten Daten übernommen, und die Hauptseite wird präsentiert.

4.5.1 Produkt aufnehmen

Neue Produkte werden einer Kategorie zugeordnet. Dazu wird auf der Hauptseite die Aktion *Produkt aufnehmen* ausgeführt, woraufhin die Anwenderin eine Kategorie auswählt. Je nach konkreten Eigenschaften der Produkte dieser Kategorie wird eine Seite konstruiert, auf der einzelne Felder bearbeitet werden können. In Abbildung 13 ist das Formular zur Bearbeitung eines Pro-

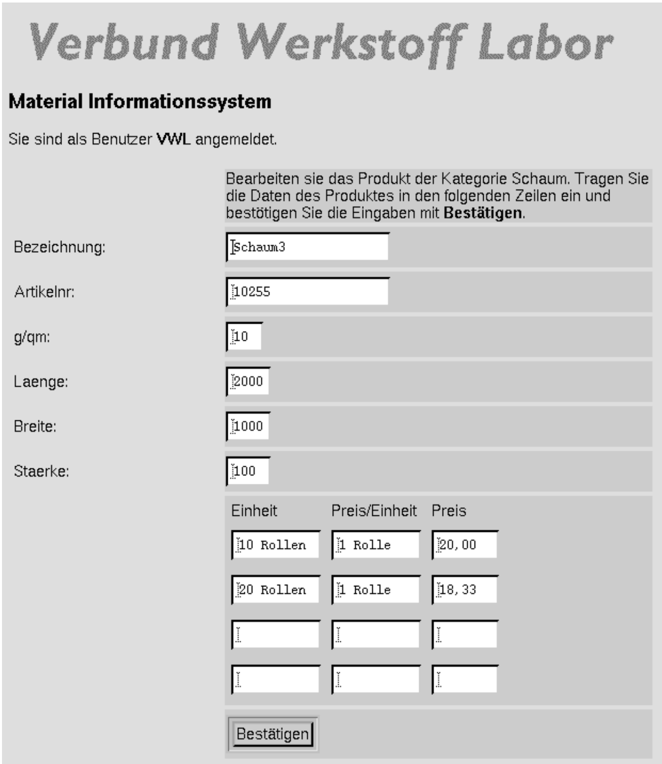

Abbildung 13: Bearbeitung eines einzelnen Produktes

duktes der Kategorie Schaum zu sehen. Wie alle Produkte hat ein Schaum beispielsweise eine Bezeichnung und eine Artikelnummer. Darüber hinaus kann ein Schaum durch sein Gewicht (g/qm) und seine Abmaße bestimmt sein. Wie bereits in Abschnitt 4.4.3 erwähnt, können Produkte jeweils in mehreren Preis/Einheit Kombinationen erfaßt werden. Auf den Seiten zur Erfassung der Produkte stehen zunächst Felder für vier solche Kombinationen bereit. In die Datenbank übernommen werden jedoch nur diejenigen Kombinationen, deren Felder Werte enthalten. So werden dem Produkt in Abbildung 13 zwei Preis/Einheit Kombinationen zugeordnet, da die beiden unteren Zeilen nicht ausgefüllt sind. Die Daten werden durch Betätigen der Schaltfläche *Bestätigen* in die Datenbank übernommen.

4.5.2 Produkt bearbeiten

Die Seite zur Bearbeitung bereits existierender Produkte sieht ebenso aus wie die im vorigen Abschnitt beschriebene Seite zum Aufnehmen von Produkten. Um die Seite zur Bearbeitung zu erreichen, sucht die Anbieterin das entsprechende Produkt so aus dem Datenbestand, wie es eine Nutzerin der Anbieterseite tun würde. Sie führt also auf der Hauptseite die Aktion *zu den Anbietern* aus, wählt aus der Spalte *Angebot* seine Angebotsliste aus, wählt die entsprechende Kategorie und dann das Produkt. Wenn die aktuelle Anwenderin ebenfalls Anbieterin des Produktes ist, können auf der Produktseite zwei weitere Aktionen ausgeführt werden: *Produkt bearbeiten* und *Produkt entfernen* (siehe Abb. 14) Soll das Produkt bearbeitet werden, die Anwenderin also die Aktionsfläche *Produkt bearbeiten* wählt, wird wiederum die Seite in Abbildung 12 erzeugt. Sind einem Produkt mehr als zwei Einheit/Preis Kombinationen zugeordnet, werden zwei zusätzliche Felder erzeugt, in die neue Kombinationen eingetragen werden können. Mit diesem Mechanismus wird verhindert, daß die Anwenderin zu weit in der Gesamtstruktur des Systems verzweigen muß. Das wäre der Fall gewesen, wenn eine eigene Aktionsfläche *neues Kombination aufnehmen* auf eine eigene Seite für Einheit/Preis-Kombinationen führen würde.

Wird hier wiederum die Aktionsfläche *Bestätigen* gewählt, wird die Anwenderin wieder zu der Seite geführt, auf die Daten des Produktes angezeigt werden.

Abbildung 14: Die Sicht auf ein Produkt für die Anbieterin mit Aktionsfeldern zum Bearbeiten und Entfernen

4.5.3 Produktseite bearbeiten

Jeweils einem konkreten Produkt kann eine Produktseite zugeordnet sein. Diese Art Dokument ist bewußt einfach gehalten. Zwar wäre es auch möglich gewesen, der Anbieterin die Möglichkeit zu geben, ihre Seite direkt in HTML zu beschreiben. Dies ist jedoch wenig benutzungsfreundlich, und könnte potenzielle Anbieterinnen von der Nutzung des Systems abhalten. Eine Produktseite kann aus mehrern Absätzen bestehen, wovon jedem ein Bild zugeordnet sein kann. Jeweils ein Absatz wird in das Textfeld auf der linken Hälfte der Seite eingegeben während ein optinal diesem Absatz zugeordnetes Bild in dem Auswahlfeld rechts neben dem Absatz ausgewählt wird (siehe Abb. 15). Die Bilder, die ausgeählt werden können, werden zuvor ebenfalls

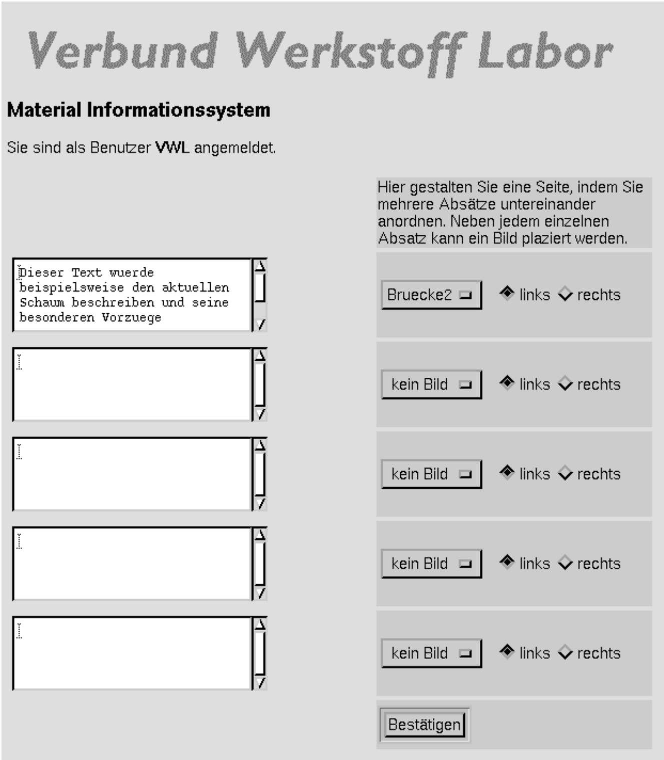

Abbildung 15: Die Ansicht zur Bearbeitung einer Produktseite

mit Hilfe einer grafischen Schnittstelle auf das System geladen und stehen nur der entsprechenden Anwenderin zur Verfügung(siehe 4.5.5. Dies ist die einzige Möglichkeit, trotz der Beschränkungen des HTTP-Protokolls einfache Dokumente auf der Client-Seite zu erzeugen.

Nachdem alle Absätze bearbeitet und eventuell mit Bildern versehen wurden, wird dieses Dokument in die Datenbank aufgenommen und kann von Anwenderinnen betrachtet werden. In der Produktansicht erscheint die Produktseite unterhalb der strukturierten Daten (siehe Abb. 12). Solch ein Dokument wird wieder nach dem gleichen Prinzip verändert, wie es schon bei der Bearbeitung der Produktdaten war: Zuerst wird die entsprechende Seite gesucht, wie es auch eine fremde Anwenderin tun würde. Wenn die Anwenderin ebenfalls Anbieterein des ausgeählten Produktes ist, erscheint eine zusätzliche Aktionsfäche *Produktseite bearbeiten*, über die das Dokument bearbeitet werden kann.

4.5.4 Anbieterseite bearbeiten

Ebenso wie eine Produktseite kann auch die Seite einer Anbieterin bearbeitet werden. Auch hier werden einzelne Absätze mit Bildern kombiniert. Das Ergebnis einer solchen Bearbeitung wird in die Datenbank aufgenommen und wird angezeigt, wenn die Anwenderin wie in Abschnitt 4.4.1 beschrieben eine Anbieterin auswählt.

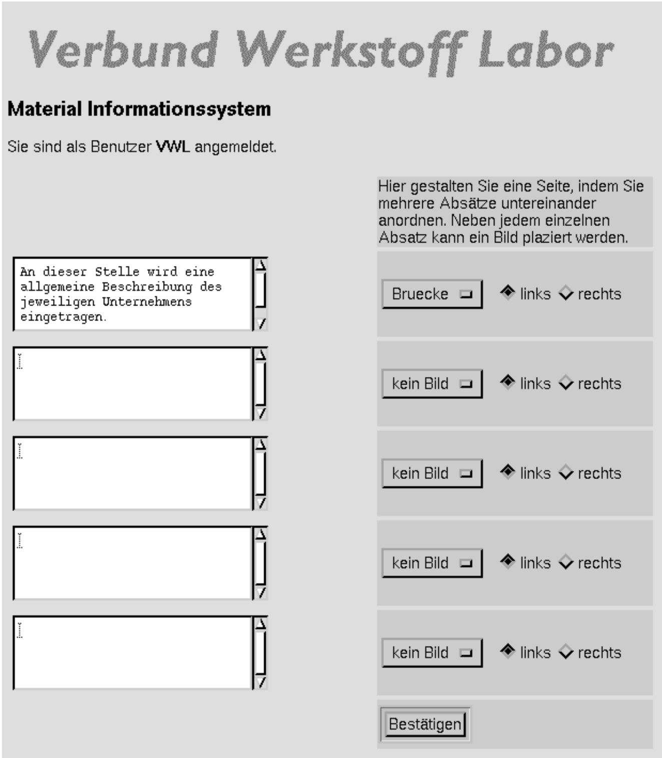

Abbildung 16: Die Ansicht zur Bearbeitung einer Anbieterseite

4.5.5 Bilder aufnehmen

Bilder, die in Produktseiten oder Präsentationsseiten für Anbieterinnen auftauchen, werden über eine grafisches Schnittstelle aufgenommen. Führt die Anbieterin die Aktion *Bilderverwaltung* auf der Hauptseite aus, kann sie Bilder von ihrem eigenen System auf den Server laden. Dort kann eine *gif* oder jpg Datei ausgewählt werden. Diese wird mit einem Titel versehen und mit der Aktion *Abschicken* an den Server übergeben. Im folgenden kann die Anbieterin dieses Bild beispielsweise bei der Gestaltung der Produktseiten auswählen (siehe Abb. 17).

Abbildung 17: Die Ansicht zum Aufnehmen von Bildern

5 Zusammenfassung

Mit dem Material-Informationssystem Verbundwerkstoffe ist ein Internet-basiertes System entstanden, in dem die Komplexität einer datenbankgestützten Anwendung vor den Benutzerinnen verborgen wird. Insbesondere die Nutzung einer WWW-Schnittstelle erleichtert den Zugang zum System, da auf den Rechnern der Benutzerinnen keine eigenständige Anwendung installiert werden muß.

Literatur

[Ame92] American National Standards Institute. *American national standard for information systems: database language — SQL: ANSI X3.135-1992.* American National Standards Institute, 1430 Broadway, New York, NY 10018, USA, 1992.

[BCN92] Carlo Batini, Stefano Ceri, and Shamkant Navathe. *Conceptual Database Design- an Entity-Relationship Approach.* Benjamin/Cummings, Redwood City, 1992.

[bre] Die Bremer Stadtinformanten, `http://www.bremen.de/`.

[CBB+97] R. G. G. Cattell, D. Barry, D. Bartels, M. Berler, J. Eastman, S. Gamerman, D. Jordan, A. Springer, H. Strickland, and D. Wade. *The Object Database Standard: ODMG 2.0.* Morgan Kaufmann Publishers, Los Altos (CA), USA, 1997.

[CCD+99] Stefano Ceri, Sara Comai, Ernesto Damiani, Piero Fraternali, Stefano Paraboschi, and Letizia Tanca. XML-GL: A graphical language for querying and restructuring XML documents. In *Proceedings of the Eighth World-Wide Web Conference*, 1999.

[Che83] P. P. Chen, editor. *Entity-Relationship Approach to Information Modeling and Analysis*, Amsterdam, 1983.

[Cor98] Barry Cornelius. Using CORBA and JDBC to produce three-tier systems. *ACM SIGPLAN Notices*, 33(4):44–52, April 1998.

[CS95] Tiziana Catarci and Giuseppe Santucci. Are visual query languages easier to use than traditional ones? an experimental proof. In *Proceedings of the HCI'95 Conference on People and Computers X*, Visualisation, pages 323–338, 1995.

[DFMR95] F. Di Loreto, F. Ferri, F. Massari, and M. Rafanelli. A visual object-oriented query language for geographic information systems. *Lecture Notes in Computer Science*, 978:103–??, 1995.

[GH91] Martin Gogolla and Uwe Hohenstein. Towards a semantic view of an extended entity-relationship model. *ACM Transactions on Database Systems*, 16(3):369–416, September 1991.

[OMG99] OMG, editor. *OMG Unified Modeling Language Specification, Version 1.3, June 1999.* Object Management Group, Inc., Framingham, Mass., Internet: `http://www.omg.org`, 1999.

[RGB99] Oliver Radfelder, Martin Gogolla, and Volker Behr. On Designing and Querying a Material Information System. In M. Türksever, N.Y. Topaloglu, N. Zincir-Heywood, A. Kantarci, C.A. Sürgevil, and Ö. Özmen, editors, *Proc. 14th Int. Symposium Computer and Information Science (ISCIS'99)*, pages 132–142. Ege University, Izmir, 1999.

[Shn94] Ben Shneiderman. Dynamic queries for visual information seeking. Technical Report UMCP-CSD CS-TR-3022, Department of Computer Science, University of Maryland, College Park, Maryland 20742, U.S.A., January 1994. original version: September1993.

[Vee95] Aravindan Veerasamy. Querying, navigating and visualizing an online library catalog. In *Proceedings of the Eighteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Systems Demonstrations: Abstracts, page 367, 1995.

[WM97] Alan R. Williamson and Ceri L. Moran. *Java Database Programming: Servlets & JDBC.* Prentice Hall, 1997.