Universität Hamburg
Fachbereich Informatik
Arbeitsbereich Datenbanken und Informationssysteme

Studienarbeit

Objektorientierte Analyse und Entwurf eines Internet-Produkt-Informationssystems

eingereicht bei:

Prof. Dr. Florian Matthes
Arbeitsbereich Softwaresysteme

vorgelegt von:

Lan Zhang
Am Centrumshaus 2
21073 Hamburg
Tel: 040 – 7654811

Hamburg, den 15.Juli 1998

Zusammenfassung

Die vorliegende Arbeit beschreibt die Analyse und den Entwurf eines Produktinformationssystems - PIA. Dabei wird ein objektorientierte Ansatz der Softwareentwicklung verfolgt, der aus den drei Phasen Analyse, Entwurf und Implementierung besteht.

Im 2. Kapitel dieser Arbeit wird ein Analysemodell entwickelt, in dem zuerst die Anwendungsfälle des Systems beschrieben werden, um grundlegende Aspekte der Anwendungsdomäne ungeachtet der späteren Implementierung zu identifizieren und zu beschreiben. Innerhalb dieser Analysephase werden der Anforderungskatalog und das Use-Case-Modell erstellt, sowie die möglichen Akteure des Systems herausgearbeitet.

Im 3. Kapitel wird auf der Basis dieses Analysemodells ein Objektmodell aufgebaut, das Objekte enthält, die sich aus der Anwendungsdomäne herauskristallisiert haben, sowie eine Beschreibung ihrer Eigenschaften durch Attribute und ihres Verhaltens durch Operationen, die auf diese Objekte angewendet werden können. Im zweiten Teil des 3. Kapitels werden die Interfaceklassen, die zur Darstellung der Informationen und Interaktion mit dem Anwender dienen, identifiziert und ihre Beziehung mit den Objektklassen, die in dem ersten Teil des 3. Kapitels vorgestellt werden, durch Interaktionsdiagramme exemplarisch dargestellt.

Inhaltsverzeichnis

Abbildungsverzeichnis **3**
Kapitel 1 Einleitung **4**
1.1 Gliederung der Arbeit 5
Kapitel 2 Analyse **7**
2.1 Systemüberblick 7
2.2 Das Vokabular 9
2.3 Anwendungsfälle und Entwurf der Benutzerschnittstelle 10
2.3.1 Anwendungsfälle der Kunden 10
2.3.1.1 Benutzeranmeldung und –Registrierung 11
2.3.1.2 Erstellung neuer Suchaufträge 12
2.3.1.2.1 Anwendungsfall Attributanfragen 13
2.3.1.2.1.1 Skalare Anfrage 14
2.3.1.2.1.2 Hierarchische Anfrage 15
2.3.1.2.1.3 Aufzählungsanfrage 16
2.3.1.2.1.4 Volltextanfrage 16
2.3.1.2.2 Ergebnis anzeigen 17
2.3.1.2.3 Ergebnis formatieren 17
2.3.1.2.4 Suchaufträge speichern/abonnieren 18
2.3.1.3 Anwendungsfall Bearbeitung Suchaufträge 19
2.3.2 Weitere Anwendungsfälle für Anbieter 20
2.3.2.1 Produkte einfügen 21
2.3.2.2 Produkte ändern/löschen 21
2.3.3 Anwendungsfälle für Redakteur 23
2.3.3.1 Domänen 23
2.3.3.1.1 Skalar 23
2.3.3.1.2 Hierarchische Domäne 24
2.3.3.1.3 Aufzählungsdomäne 24
2.3.3.1.4 Volltextdomäne 24
2.3.3.2 Verwaltung Attribute/Domäne 24
2.3.3.2.1 Einfügen Attribute/Domänen 24
2.3.3.2.1.1 Einfügen skalarer Attribute/Domänen 25
2.3.3.2.1.2 Einfügen hierarchischer Attribute/Domäne 25
2.3.3.2.1.3 Einfügen von Aufzählungsattribute/-domänen 26
2.3.3.2.1.4 Einfügen eines Volltextattributes 26
2.3.3.2.2 Löschen Attribute/Domänen 27
2.3.3.2.3 Ändern Attribute/Domänen 27
2.3.3.2.3.1 Ändern eines skalares Attribut/einer skalarer Domäne 27
2.3.3.2.3.2 Ändern eines hierarchisches Attributes/einer hierarchischen Domäne 27
2.3.3.2.3.3 Ändern eines Aufzählungsattributs/einer Aufzählungsdomäne 27
2.3.3.2.3.4 Ändern eines Volltextattributs/einer Volltextdomäne 27
2.3.4 Anwendungsfälle für Administrator 27
2.3.4.1 Neue Benutzer einfügen 28
2.3.4.2 Benutzer ändern/löschen 28
2.4 Zusammenfassung der Analysephase 28
Kapitel 3 Objektmodellierung **29**
3.1 Die Objekt-Klassen 30
3.1.1 Die Klassen Akteur, Rolle 32
3.1.2 Die Klassen Kunde, Suchauftrag 33
3.1.3 Die Klasse Suchauftrag 35
3.1.4 Die Klasse Attributanfrage 36
3.1.5 Das Klassendiagramm von Redakteur, Attribut, Domäne 36
3.1.6 die Klassen Anbieter, Produkt usw 39
3.1.7 Die Klasse Anbieter 39
3.1.8 Die Klasse Produkt und die Klasse Treffer 40
3.2 Die Controlklasse PIA 40
3.3 Java Interface Klassen 41
3.3.1 Warum Java 41

3.3.2 Was ist Java ... 42
3.3.3 Interface Klassen ... 42
3.3.3.1 Interfaceklassen in Überblick ... 43
3.3.3.1.1 AA-Darsteller: ... 44
3.3.3.3.2 SuchauftragDarsteller ... 44
3.3.3.3.3 SuchaufträgListenDarsteller ... 45
3.3.3.3.4 AA-Editor: ... 45
3.3.3.3.5 TrefferDarsteller ... 45
3.3.3.3.6 ErgebnisDarsteller ... 45
3.3.3.3.7 AttributListeDarsteller ... 45
3.4 Interaktionen zwischen den Objektklassen und Interfaceklassen ... 45
3.4.1 Das Muster Model-View-Controller (MVC) ... 45
Kapitel 4 Zusammenfassung und Ausblick ... 49
Literaturverzeichnis ... 512

Abbildungsverzeichnis

Abbildung 1.1: Benutzerorientiertes Informationssystem *4*
Abbildung 2.1: PIA Systemüberblick *8*
Abbildung 2.2: Kunden-Anwendungsfälle in Überblick *11*
Abbildung 2.3: Zustandsdiagramm Benutzeranmeldung/-registrierung *12*
Abbildung 2.4: Ablauf Erstellung neuer Suchaufträge *12*
Abbildung 2.5: Zustandsdiagramm Attributanfragen *13*
Abbildung 2.6: skalare Anfrage mit einer Zielgröße *14*
Abbildung 2.7: skalare Anfrage mit einem Zielbereich *14*
Abbildung 2.8: Benutzerschnittstelle für hierarchische Anfrage *15*
Abbildung 2.9: Zustandsdiagramm für den Anwendungsfall Attributanfrage *17*
Abbildung 2.10: Suchergebnis anzeigen *17*
Abbildung 2.11: Ergebnis Formatierung *18*
Abbildung 2.12: Benutzerschnittstelle für den Anwendungsfall Bearbeitung Suchaufträge *19*
Abbildung 2.13: Zustandsdiagramm Bearbeitung Suchaufträge *20*
Abbildung 2.14: Zustandsdiagramm des Anwendungsfalls Anbieteranmeldung *21*
Abbildung 2.15: Zustandsdiagramm Produktpflege *23*
Abbildung 2.16: Einfügen eines Attributes *25*
Abbildung 2.17: Einfügen Hierarchischer Domäne *26*
Abbildung 2.18: Einfügen Aufzählungs-Domäne *26*
Abbildung 3.1: PIA Klassen in Überblick *31*
Abbildung 3.2: Die Klassen Akteur, Rolle *32*
Abbildung 3.3: Die Klasse Dictionary und die Klasse List *33*
Abbildung 3.4: Die Klassen Kunde, Suchauftrag, Attributanfrage *35*
Abbildung 3.5: Die Klassen Redakteur, Domäne usw. *37*
Abbildung 3.6: Anbieter, Produkt usw. *39*
Abbildung 3.7: Interfaceklassen für Erstellung eines neuen Suchauftrages *43*
Abbildung 3.8: Interfaceklassen in Überblick *44*
Abbildung 3.9: Interaktionsdiagramm zwischen den Objekten *447*

Kapitel 1

Einleitung

Seitdem das WWW 1993 unter Federführung des CERN als Informationssystem für die Hochenergie Physik vorgestellt wurde, hat es diesen engen Rahmen längst gesprengt und ist zum Selbstläufer geworden. Viele Unternehmen sind als Informationsanbieter im WWW präsent. Es wird jedoch immer schwieriger, die unzähligen Informationen des WWW für eine gezielte Suche aufzubereiten. Daher werden viele Informationssysteme im Internet angeboten. Ein Informationssystem kann als ein System beschrieben werden, das aus einer Menge von Suchanfragen, einer Menge von Datensätzen und einem Mechanismus besteht, der die für eine Suchanfrage relevanten Datensätze bestimmt. Ein Hauptprinzip beim Design eines Informationssystems ist, daß sich das System an den Informationsbedarf der Benutzer des Systems anpassen soll. Die Benutzer eines Informationssystems (vgl. Abb. 1.1) können in zwei Gruppen unterteilt werden:

- Informationsanbieter und
- Informationsnutzer.

Ein benutzerorientiertes Informationssystem stellt als ein Brokerdienst eine gemeinsame Schnittstelle für seine Benutzer zur Verfügung, wo nicht nur Informationsangebote von einem Informationsanbieter aufgenommen werden, sondern auch Suchanfragen eines Informationsnutzers bearbeitet werden und dadurch dessen speziellen Informationsbedarf befriedigt werden kann.

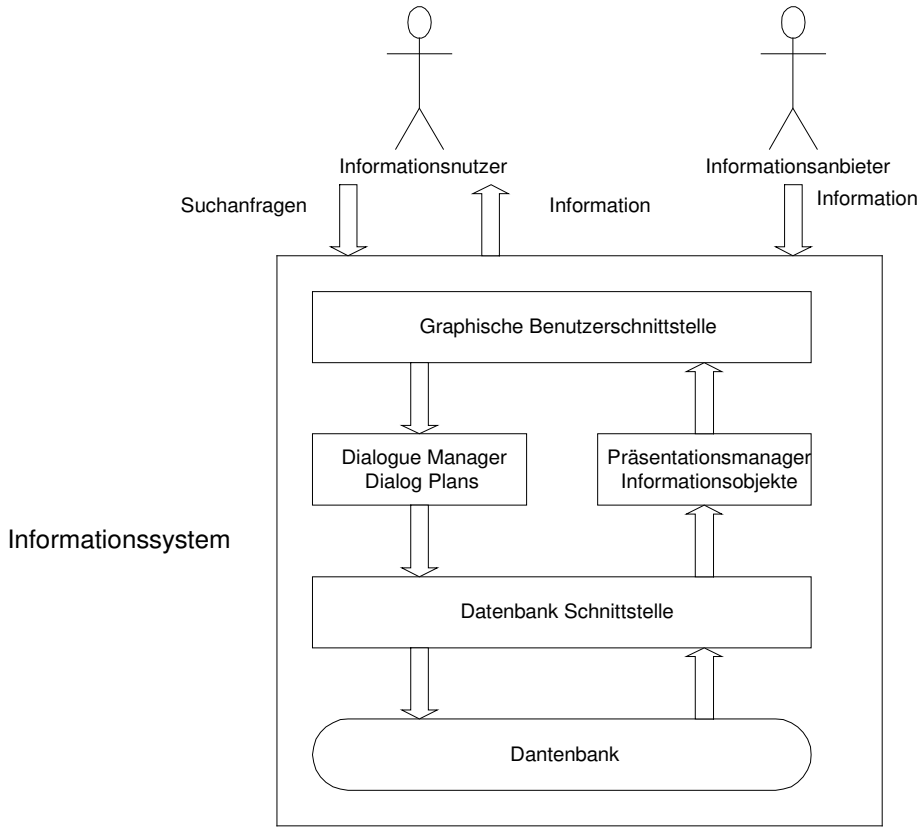

Abbildung 1.1: Benutzerorientiertes Informationssystem

Ein erfolgreiches benutzerorientiertes Informationssystem im Internet zeichnet sich dadurch aus, daß es

- die persönlichen Bedürfnisse jedes Benutzeres als Individuum berücksichtigt,
- die passenden Produkte oder Dienste aus einer großer Produkt- oder Dienstdatenbank identifiziert,
- einen intelligenten, benutzerorientierten Suchprozeß unterstützt, mit dem ein Benutzer die Produkte und Dienste für seine persönlichen Bedürfnisse schnell finden kann.

Typische Beispiele der Interaktionsprozesse mit den Internet-Suchdiensten sind:

1. In einem Produktkatalog eines Online-Shopping-Systemes suchen nach „dem billigsten VCR mit Stereo Sound und Jog Schuttle“.
2. Suchen in den klassifizierten Werbungen in einer Online-Zeitung nach „einem guten chinesischen Restaurant, das Mastercard akzeptiert“.
3. In einem Dienstverzeichnis suchen nach
   - Finanzendiensten wie „einer Autoversicherung für ein BMW mit mehr als zwei Fahrern“;
   - Wartungsdiensten in der Seeindustrie wie „einem Schrauberhersteller für Titaniumstahl in Europa“.
4. Bibliographisches Suchen in einer digitaler Bibliothek nach „einem Zeitschriftartikel von F. Matthes über Migrating Threads“.

Die Technologien für solche Prozesse, die z.Z. im Internet verwendet werden, sind nicht ausgereift genug, um Benutzerpräferenzen zu berücksichtigen. Bei den Volltext-Suchmaschinen wird die semantische Struktur von Produktbeschreibungen nicht berücksichtigt. Relationale Tabellen und relationale Anfragen unterstützen keine Fuzzy-Objekte und –Anfragespezifikationen. Hierarchische Klassifikationsstrukturen zwingen den Kunden einen statischen linearen Entscheidungsbaum auf.

Es besteht die Notwendigkeit, neue Informationssysteme zu schaffen, die basierend auf neuen Modellen zur Unterstützung von benutzerorientierten Geschäftsprozessen rapiden und kontinuierlichen Veränderungen gewachsen sind.

Ziel dieser Arbeit ist die Analyse und Entwurf eines Daten- und Suchmodells – PIA (Product Information Assistant), das

- Fuzzy-Suchspezifikationen unterstützt,
- sortierte, heterogene Suchergebnisse liefert und
- eine benutzerfreundliches, interaktive grafische Benutzerschnittstelle besitzt.

1.1 Gliederung der Arbeit

In dieser Arbeit wird ein objektorientierter Entwicklungsansatz verfolgt, der sich weitestgehend an dem Vorgehensmodell der Unified Modelling Language (UML) anlehnt. Nach diesem Modell wird der Entwicklungsprozeß in drei Phasen untergliedert – Analyse, Entwurf und Implementierung, wobei der Schwerpunkt dieser Arbeit auf der Analyse- und der Entwurfsphase liegt.

Im Rahmen der vorliegenden Arbeit werden folgende Punkte ausgeführt:

- In 2. Kapitel wird ein Analysemodell aufgebaut, um grundlegende Aspekte der Anwendungsdomäne ungeachtet der späteren Implementierung zu identifizieren und zu beschreiben. Innerhalb dieser Analysephase werden der Anforderungskatalog und das Use-Case-Modell erstellt, sowie die möglichen Akteure des Systems herausgearbeitet.

- Im 3. Kapitel wird ein objektorientiertes Entwurfsmodell vorgestellt, das Objekte enthält, die sich aus der Anwendungsdomäne herauskristallisiert haben, sowie eine Beschreibung ihrer Eigenschaften durch Attribute und ihres Verhaltens durch Operationen, die auf diese Objekte angewendet werden können. Im zweiten Teil des 3. Kapitels werden die Interfaceklassen, die zur Darstellung der Informationen und Interaktion mit dem Anwender dienen, identifiziert und ihre Beziehung mit den Objektklassen, die in dem ersten Teil des 3. Kapitels vorgestellt werden, durch Interaktionsdiagramme exemplarisch dargestellt.

Kapitel 2

Analyse

In diesem Kapitel wird ein Anwendungsfallsmodell entwickelt, um die Anforderungen an das System zu modellieren und die erforderlichen Funktionalitäten des Systems zu beschreiben. Um die Anforderungsanalyse des Systems vorzunehmen, ist zuerst festzustellen, welche Akteure an dem System beteiligt sind und welche Funktionalitäten sie benötigen. Im Systemüberblick werden die gesamten Systemanforderungen grob dargestellt, während die Handhabung dieser Funktionalitäten in einem späteren Abschnitt detaillierter beschrieben wird.

2.1 Systemüberblick

PIA steht für „Product Information Assistant“ und stellt einen Brokerdienst zwischen Informationsanbietern und Informationsnutzern dar. PIA verfügt über eine Datenbank, in der Informationen über viele Produkte gespeichert werden. Daraufhin stellt PIA eine gemeinsame Schnittstelle für seine Benutzer zur Verfügung, die es ihnen ermöglicht, Suchanfragen nach bestimmten Produkten durchzuführen bzw. neue Produkte ins System aufzunehmen. Dementsprechend werden in zwei Arten von Benutzern unterschieden (vgl. Abb. 2.1):

- Kunde und
- Anbieter.

Unter „Kunde“ verstehen wir im folgenden die Benutzer, die Suchanfragen nach bestimmten Produkten an das System stellen, während ein Anbieter außerdem seine Produkte einfügen, ändern und löschen kann.

Außerdem gibt es zwei weitere Benutzergruppen, die Administrationstätigkeiten ausführen:

- Redakteur
- Administrator.

Der Redakteur ist für die Verwaltung der gesamten Attribute und Domänen zuständig (vgl. Abschnitt), während sich der Administrator in erster Linie um die Benutzerverwaltung kümmert.

Das PIA System soll folgende Merkmale besitzen:

- Interaktivität,
- Universität,
- Generizität,
- Benutzerzugangskontrolle, die zwischen den verschiedenen Benutzergruppen unterscheidet,
- Unterstützung der Fuzzy Logik.

Interaktivität bedeutet, daß das System aus der Kundensicht in der Lage sein soll, eine Suchanfrage an die Datenbank mit den von dem Kunden ausgewählten Attributen durchzuführen. Dabei soll der Kunde nicht mit der Anfragesyntax vertraut sein müssen, sondern viel mehr die Anfrage dadurch erstellen, daß er die für ihn wichtigen Attribute aus dem Dialog des Systems aussucht, und ggf. mit Werten belegt. Außerdem sollen

benutzerdefinierte, speicherbare Einstellungen möglich sein, u.a. zum Ausgabeformat der Ergebnisliste, Benachrichtigung über neue Produkte etc.

Das PIA System soll universell sein, d.h., daß PIA aus der Sicht eines Anbieters jedes seiner derzeit vorhandenen sowie auch alle zukünftigen Produkte über das System anbieten kann. Dabei sollte es keine Einschränkungen bezüglich der nutzbaren Attribute geben. Der Anbieter muß in der Lage sein, für jedes seiner Produkte alle seiner Meinung nach wichtigen Daten anzugeben, so daß kein Informations- und Attraktivitätsverlust auftritt.

Zugleich soll PIA generisch sein. Der Redakteur ist dafür zuständig, die Attributliste und die Domäne zu verwalten. Die Domäne enthält u.a. die Informationen über die vorhandenen Datentypen, ihre Einheiten sowie Formeln zur Umrechnung anderer Einheiten auf die normierten Standardeinheiten. Für ihn ist es wichtig, daß keine Änderung an der Attributliste oder Domäne, egal welcher Art, eine Veränderung der Benutzerschnittstelle erfordert. Das System aktualisiert ggf. die Dialoge selbständig.

PIA verfügt eine Zugangskontrolle. Jemand, der mit PIA arbeiten möchte, muß zuerst einen Benutzerantrag stellen. Er erhält einen Benutzernamen von PIA und wird einer bestimmten Benutzergruppe zugeordnet. Damit sind die Funktionalitäten von PIA, die er in Anspruch nehmen kann, eindeutig festgelegt.

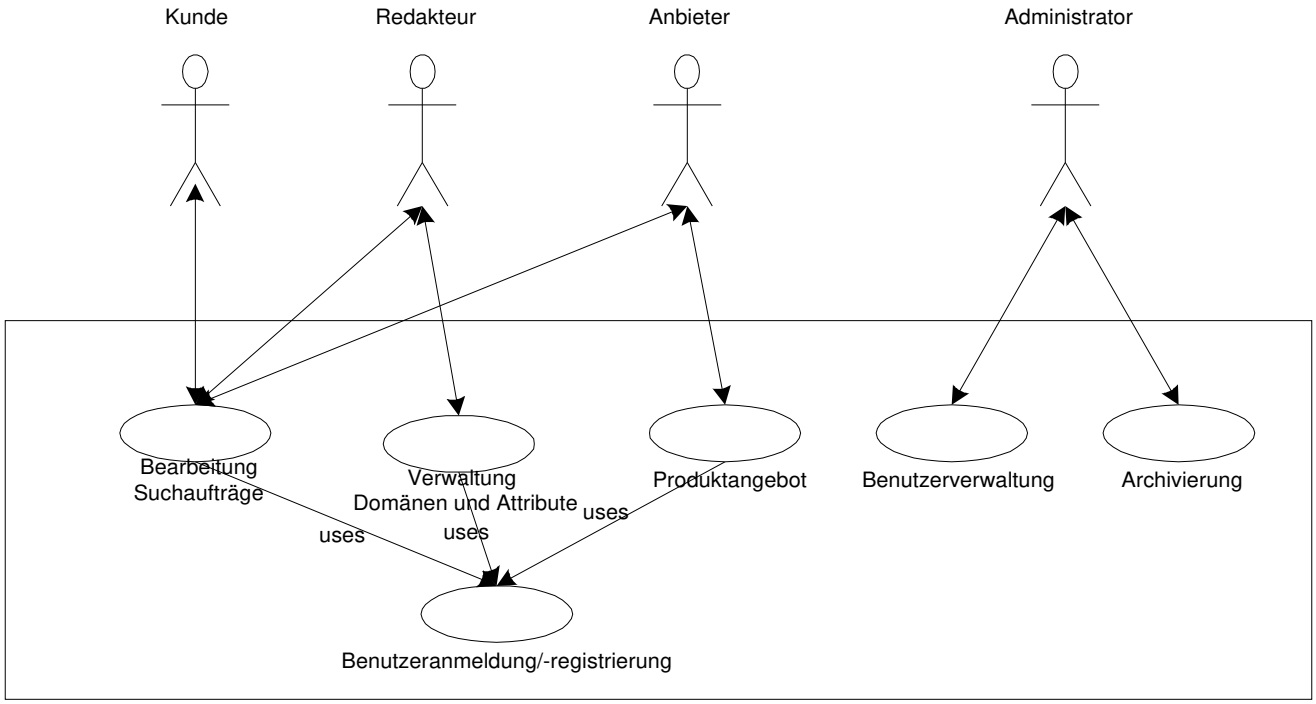

Abbildung 2.1: PIA Systemüberblick

Um die Objekte und die Ablaufprozesse des realen Systems herauszufinden und zu identifizieren, werden die Anwendungsfälle betrachtet und analysiert. Dazu wird ein wohldefiniertes Vokabular benutzt, das im folgenden vorgestellt wird.

2.2 Das Vokabular

Produkt:

Ein Produkt ist ein verkaufbarer Artikel und wird einer bestimmten Produktkategorie zugeordnet. Es hat weitere attributbezogenen Beschreibungen wie einen bestimmten Namen oder auch eine Preisangabe sowie produktspezifische Attribute. Ein Beispiel für ein Produkt ist:

Kategorie Auto, Typ Mercedes E230, Baujahr 1998, Leistung 1000 PS, Preis 99,00 DM.

Attribut:

Synonym: Merkmal, Eigenschaft

Eine mit einem Namen (Bezeichner) versehende Eigenschaft eines Produkts.

Attribute sind sowohl informationsseitig betrachteten Entitäten (in der realen Welt) als auch datenseitig betrachteten Entitäten zugeordnet. Meistens kommt ihre Bedeutung in ihrem Namen zum Ausdruck.

Beispielsweise informieren die Attribute „Produktname“, „Produktkategorie“, „Preis“, „Hersteller“ und „Anbieter“ über den Zustand eines Produktes (in der realen Welt) sowie über die Bedeutungen der Ausprägungen entsprechend benannten Datenfelder in Datensätzen einer Produktdatei.

Ausprägung:

Synonym: Attributwert

Ein konkretes Vorkommen (tatsächliches Auftreten) eines Attributes.

In aller Regel ist nicht die Gesamtmenge aller denkbaren Attributwerte von Interesse, sondern nur eine von der jeweils in Betracht gezogenen Objektklasse bestimmten und abgegrenzte Teilmenge, die ggf. als Wertbereich zu vereinbaren ist.

Beispiel: Im Produktkatalog wird die Produktkategorie „Personal Computer“ genannt, Attribut ist „Produktkategorie“, Attributausprägung ist „Personal Computer“.

In einer tabellarischen Darstellung werden Attributausprägungen als Tabelleneinträge dargestellt, während ein Attribut durch die Spaltenüberschrift einer Tabelle repräsentiert wird.

Domäne:

Jedem Attribut ist eine Domäne zugeordnet. Die Domäne enthält die Information über die gesamte Menge der Attributausprägungen, die bei diesem Attribut vorkommen können.

Ein Beispiel dafür ist die Domäne „Preis“, die positive reelle Zahlen repräsentiert und weitere Informationen zu Währungseinheiten, z.B. Dollar, DM oder Yen usw. enthält.

Entität:

Synonym: Objekt

Eine Repräsentation eines konkreten oder abstrakten Gegenstandes, die für ein gegebenes Anwendungssystem von Bedeutung ist. Ein Produkt ist z.B. eine Entität.

Attributanfrage:

Eine Attributanfrage gibt die Ausprägungen eines Attributes wieder, nach denen die Produkte gewählt werden sollen. Wir bezeichnen z.B. „Kategorie ist Auto“ als eine Attributanfrage.

Suchauftrag:

Ein Suchauftrag besteht aus einer oder mehreren Attributanfragen, die sich auf unterschiedliche Attribute beziehen. Ein Beispiel dafür wäre:
Kategorie ist Auto, Preis zwischen 2000 DM und 10.000 DM, Leistung mindestens 100 PS.

2.3 Anwendungsfälle und Entwurf der Benutzerschnittstelle

Im folgenden werden die Anwendungsfälle, die in der Abbildung 2.2 aufgeführt wurden, detailliert beschrieben.

Jeder Anwendungsfall beschreibt einen Dienst, den der Anwender von dem System fordert. Er beinhaltet die ganze Abfolge von Interaktionen zwischen dem Anwender und dem System, die erforderlich sind, um eine Arbeit durchzuführen, die aus der Sicht des Anwenders als Ganzes von Bedeutung ist. Anwendungsfälle helfen dem Entwickler, die Systemanforderungen besser zu verstehen. Sie werden daher in Form von Szenarios aufgeschrieben. Die Anwendungsfälle werden im folgenden nach den Benutzergruppen getrennt betrachtet.

2.3.1 Anwendungsfälle der Kunden

Bevor ein Benutzer mit PIA arbeiten kann, muß er sich am System anmelden. Ein neuer Benutzer muß einen Zugang zum System beantragen. Nach der erfolgreichen Anmeldung kann ein Benutzer Suchanfragen an PIA stellen, sie speichern lassen und jederzeit auf sie zugreifen und ändern bzw. löschen. Im folgenden Diagramm wird der Ablauf illustriert.

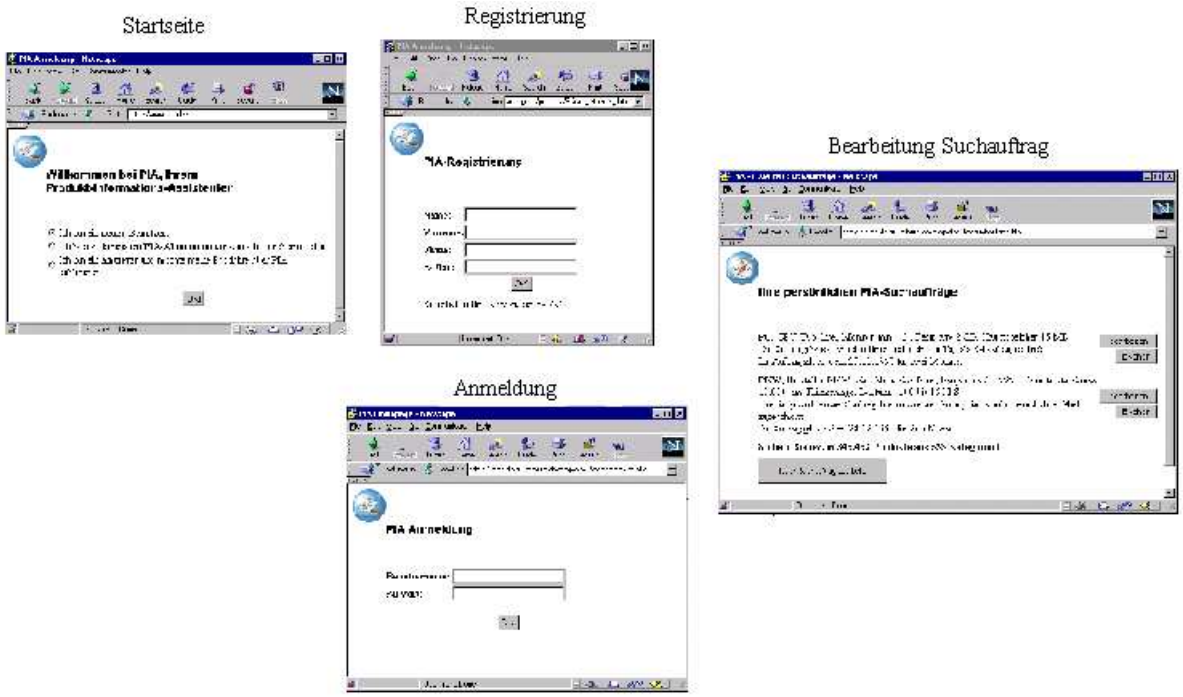

Abbildung 2.2: Kunden-Anwendungsfälle in Überblick

Demzufolge sind die wichtigsten Anwendungsfälle der Kunden die folgenden:

- Benutzeranmeldung,
- Benutzerregistrierung,
- Erstellung neuer Suchaufträge sowie
- Bearbeitung gespeicherter Suchaufträge,

die im folgenden detaillierter beschrieben werden.

2.3.1.1 Benutzeranmeldung und –Registrierung

Um zu gewährleisten, daß jeder Benutzer von PIA als ein Individuum identifiziert wird und damit er seine persönlichen Einstellungen sichern kann, verlangt PIA, daß sich jeder Benutzer anmeldet, bevor er mit PIA arbeiten kann. Durch die Eingabe des Benutzernamen und Paßwortes in das Anmeldungsformular, kann sich ein Benutzer am System anmelden. Dieses Szenario wird als „Benutzeranmeldung“ bezeichnet.

Ist der Benutzer dem System noch nicht bekannt, kann er einen Benutzerantrag stellen und sich registrieren lassen, indem er das Registrierungsformular mit Informationen füllt und über den Weg des Internets abschickt. Er bekommt eine Bestätigung von PIA, daß sein Antrag aufgenommen wurde. Via Email bekommt er später seinen Benutzernamen und Paßwort von PIA zugeschickt. Der gesamte Ablauf dieses Anwendungsfalls wird durch die Abbildung 2.3 illustriert.

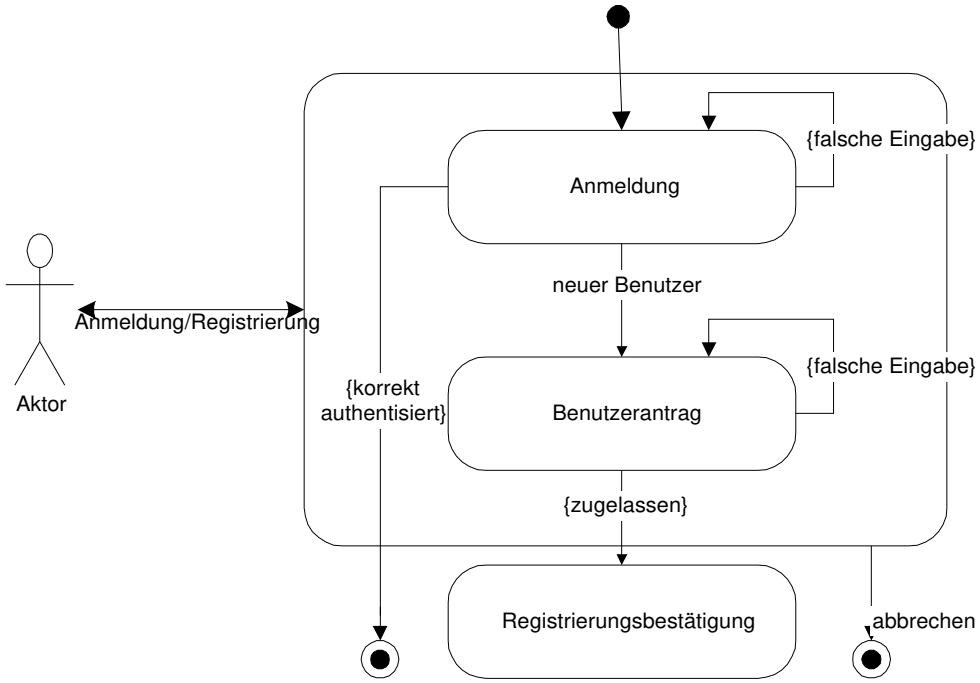

Abbildung 2.3: Zustandsdiagramm Benutzeranmeldung/-registrierung

2.3.1.2 Erstellung neuer Suchaufträge

Nach der erfolgreichen Anmeldung kann ein Kunde einen neuen Suchauftrag zusammenstellen, indem er alle Attribute, die seiner Meinung nach für die von ihm gesuchten Produkte relevant sind, wählt und deren Ausprägungen festlegt. Dieses Szenario wird als „Attributanfrage“ bezeichnet. Für den Fall, daß der Kunde schon mal mit PIA gearbeitet hat und bereits Aufträge in PIA hat speichern lassen, werden ihm nach der Anmeldung diese Suchaufträge aufgelistet, so daß er diese Suchaufträge weiter bearbeiten und ggf. löschen kann (vgl. dazu Abb. 2.4).

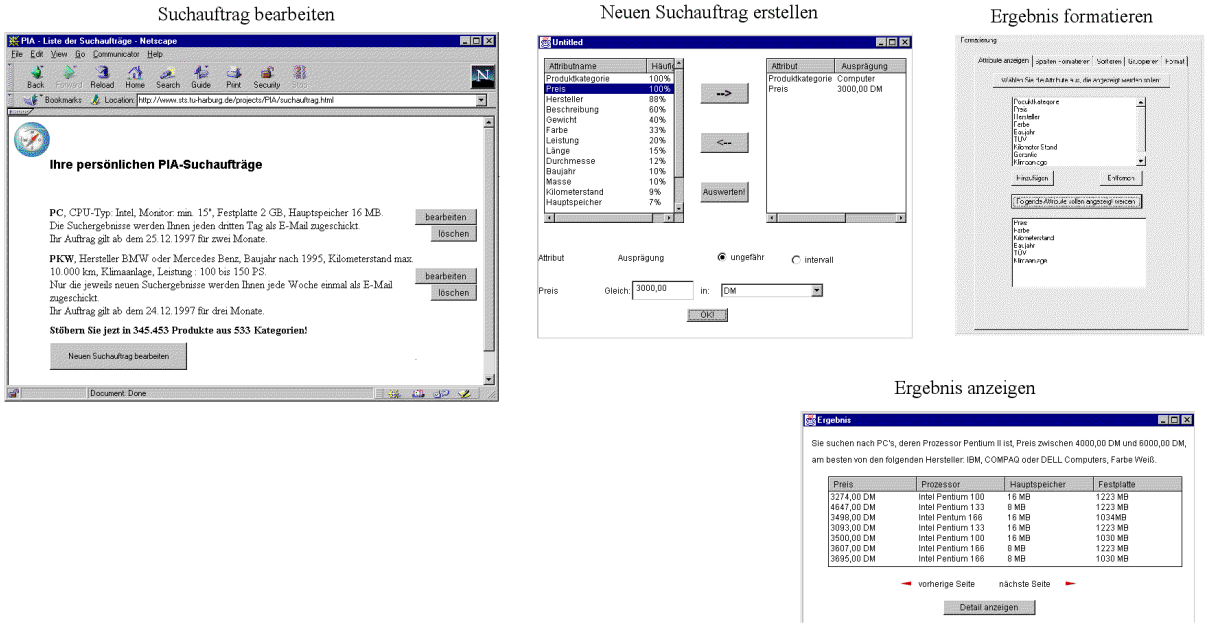

Abbildung 2.4: Ablauf Erstellung neuer Suchaufträge

2.3.1.2.1 Anwendungsfall Attributanfragen

PIA legt dem Benutzer die Wahl möglicher Attribute vor, die ihm die Produktsuche entscheidend vereinfachen kann. Der Kunde stellt seine Anfrage in Form von Wünschen, die einer Attributbeschreibung des Produktes gleichkommt, d.h. beim Zusammenstellen von Suchanfragen sucht der Benutzer die Attribute, die für ihn relevant sind, aus der Attributliste aus und belegt Sie mit den gewünschten Ausprägungen (vgl. Abb. 2.5). Jedes ausgewählte Attribut und die dazugehörigen Ausprägung(-en) bilden zusammen eine „Attributanfrage“.

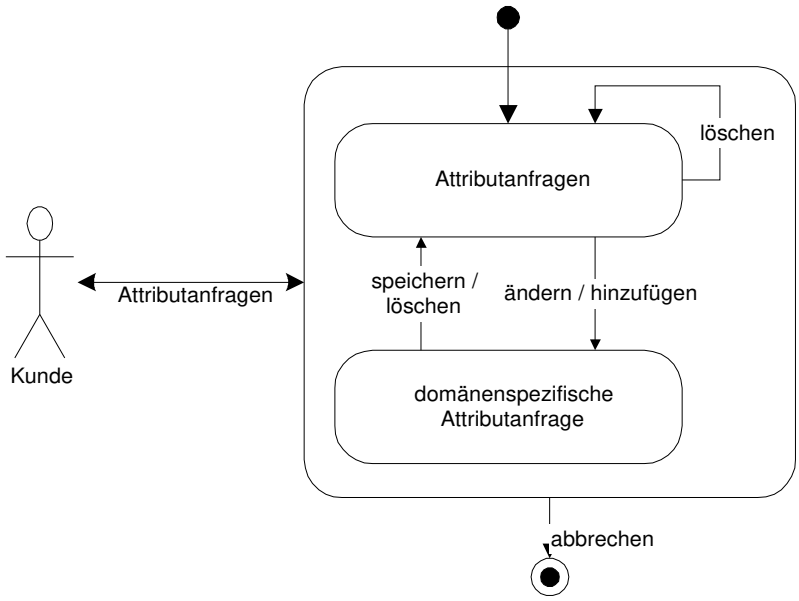

Abbildung 2.5: Zustandsdiagramm Attributanfragen

Je nachdem, was für ein Attribut der Benutzer ausgewählt hat, bietet ihm PIA eine „domänenspezifische“ Benutzerschnittstelle an, wo der Benutzer die Ausprägung(-en) festlegen kann.

In PIA ist jedem Attribut eine Domäne zugeordnet, und die Domäne enthält die Information über die gesamte Menge der Attributausprägungen, die bei diesem Attribut vorkommen können. Diese Menge kann Werte enthalten, die

- reelle Zahlen mit Maßeinheiten (z.B. 30 Kilometer) sind oder
- sich in eine Hierarchie einordnen lassen oder
- aufzählbar oder
- vom Volltexttyp[1] sind.

Entsprechend werden die Domänen in vier Unterdomänen[2] unterteilt:

- Skalar,
- hierarchische Domäne,
- Aufzählungsdomäne und
- Volltextdomäne.

[1] Volltexttyp: String
[2] Vgl. Abschnitt 2.5. Verwaltung Domänen

Für jede dieser Arten von Domänen wird eine domänenspezifische Benutzerschnittstelle entworfen, mit der der Benutzer im folgenden jeweils eine „skalare Anfrage“, „hierarchische Anfrage“, „Aufzählungsanfrage“ oder „Volltextanfrage“ formulieren kann.

2.3.1.2.1.1 Skalare Anfrage

Hat der Benutzer ein Attribut ausgesucht, dessen mögliche Ausprägungen reelle Zahlen mit Maßeinheiten sind, handelt es sich hierbei um eine skalare Anfrage. Ein solches Attribut wird als ein „skalares Attribut“ bezeichnet.

Für ein solches Attribut kann der Benutzer

- entweder eine reelle Zahl mit einer Maßeinheit angeben (die reelle Zahl mit der Maßeinheit werden zusammen als „Skalarwert“ bezeichnet), wo die gewünschte Ausprägung des Attributs ungefähr liegen soll, z.B. das Attribut „Preis“ soll ungefähr gleich 3000,00 DM sein (vgl. Abb. 2.6).

Abbildung 2.6: skalare Anfrage mit einer Zielgröße

- oder ein Intervall mit zwei Skalarwerten als untere und obere Grenze festlegen, in das die vom Benutzer gewünschte Ausprägung fallen soll z.B. „Preis“ soll zwischen „1000,00 DM“ und „2000,00“ DM sein (vgl. Abb. 2.7).

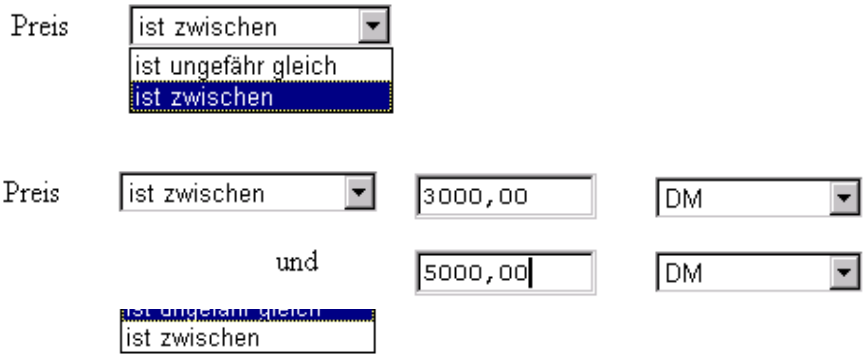

Abbildung 2.7: skalare Anfrage mit einem Zielbereich

Entsprechend sind die Operatoren für ein skalares Attribut:

- „ist ungefähr gleich“ und
- „ist zwischen ... und ...“.

Der Standardpfad dieses Anwendungsfalls ist:

1. der Benutzer wählt ein skalares Attribut aus der Attributliste aus („Attributauswahl“);

2. der Benutzer entscheidet, ob er einen ungefähren Wert oder ein Intervall angeben möchte (dieses Szenario wird als „Operatorauswahl“ bezeichnet);
3. der Benutzer gibt einen Skalarwert (für „ungefährer Wert“) oder zwei Skalarwerte (für „Intervall“) ein, nach denen die Produkte gesucht werden sollen (dieses Szenario wird als „Wertauswahl“ bezeichnet).

2.3.1.2.1.2 Hierarchische Anfrage

Wenn sich der Benutzer bei der Attributauswahl für ein Attribut entscheidet, dessen Domäne hierarchisch ist, bietet ihm PIA dann eine Benutzerschnittstelle mit einer Baumdarstellung für die Hierarchie an. Diese sieht z.B. für das Attribut „Produktkategorie“ so aus:

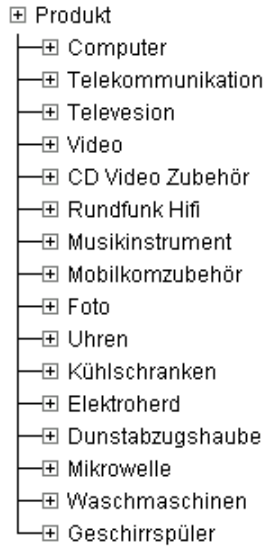

Abbildung 2.8: Benutzerschnittstelle für hierarchische Anfrage

Hier sieht der Benutzer die möglichen Produktkategorien. Durch Aufklappen der einzelnen Kategorien gelangt er zu den entsprechenden Unterkategorien. Der Benutzer sucht die gewünschte Kategorie aus und kann durch Doppelklick oder „Drag and Drop“ die Ausprägung für dieses hierarchische Attribut festlegen.

Er kann auch zugleich mehrere Produktkategorien markieren und als Ausprägungen festlegen, falls er Produkte sucht, die mehreren Kategorien untergeordnet sind.

Entsprechend sind zwei Operatoren für jedes hierarchische Attribut definiert:

- „ist gleich“ und
- „ist Teilmenge von“.

Standardpfad:

1. der Benutzer sucht ein hierarchisches Attribut aus („Attributauswahl“);
2. der Benutzer legt fest, ob die gewünschte Ausprägung von diesem Attribut ein einziger Wert oder eine Menge von Attributwerten sein soll („Operatorauswahl“);
3. der Benutzer belegt dieses Attribut mit einem Attributwert oder einer Menge von Attributwerten, die für dieses Attribut zulässig sind („Wertauswahl“);

2.3.1.2.1.3 Aufzählungsanfrage

Wählt der Benutzer ein Aufzählungsattribut aus, spricht man von einer Aufzählungsanfrage.

Die Vorgehensweise ist ähnlich wie bei einer hierarchischen Anfrage. Da es sich um eine Domäne handelt, die eine aufzählbare Menge von Werten besitzt, können in diesem Fall alle möglichen Ausprägungen von dem gewählten Attribut angezeigt werden. Dabei gibt der Benutzer an, ob er dieses Attribut mit genau einem Attributwert oder mit mehreren Attributwerten belegen möchte und wählt von den gesamten Ausprägungen diejenige(-n) aus, die seinen Vorstellungen entsprechen.

Der Standardpfad und die Operatoren sind gleich denen der hierarchischen Anfrage.

2.3.1.2.1.4 Volltextanfrage

Wählt der Benutzer ein Volltextattribut aus, handelt es sich um eine Volltextanfrage. Ein typisches Beispiel für ein Volltextattribut eines Produktes ist die Produktbeschreibung.

Hier gibt der Benutzer Text ein, den die Ausprägung des Attributes enthalten soll.

Zusammengefaßt wird der Ablauf einer „domänenspezifischen Attributanfrage“ durch das folgende Diagramm beschreiben:

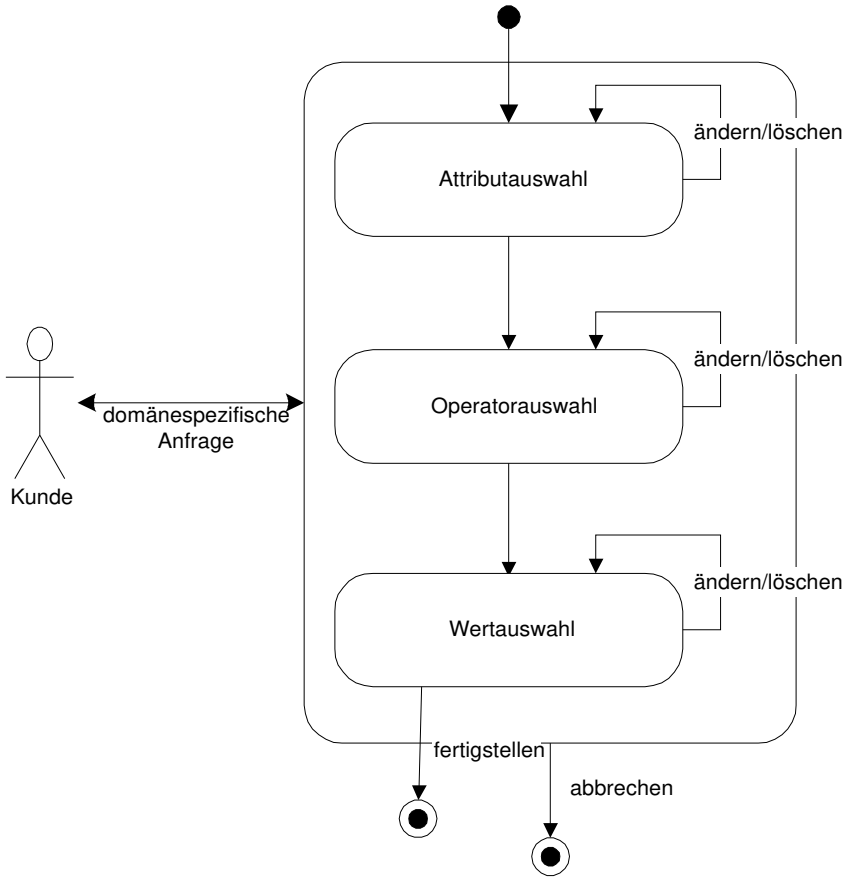

Abbildung 2.9: Zustandsdiagramm für den Anwendungsfall Attributanfrage

2.3.1.2.2 Ergebnis anzeigen

Nachdem der Benutzer einige Attribute ausgesucht und die Ausprägungen für diese Attribute festgelegt hat, kann er sie zu einer Suchanfrage zusammenstellen lassen. Die bisher ausgewählten Attribute werden zusammen mit ihren von dem Benutzer angegebenen gewünschten Ausprägungen zu einem Suchauftrag zusammengefaßt. PIA führt den Suchauftrag aus und stellt das Ergebnis dieses Suchauftrags dar. Das Suchergebnis enthält normalerweise eine Menge von Produkten. Für jedes Produkt wird eine reelle Zahl von PIA ermittelt, die besagt, wie wahrscheinlich es ist, daß ein bestimmtes Produkt für einen Suchauftrag relevant ist. Diese Zahl wird als „Relevanz“ bezeichnet. Das Produkt zusammen mit der Relevanz werden als „Treffer“ bezeichnet.

Das Suchergebnis wird standardmäßig in einem voreingestellten Format dargestellt. Es besteht aber die Möglichkeit, daß der Benutzer das Ergebnisformat individuell einstellen kann. In dem voreingestellten Format werden die Treffer in einer Tabelle nach der Relevanz absteigend aufgelistet. Das heißt, die Produkte, die die Suchanfrage am besten erfüllt haben, erscheinen auch an den ersten Positionen. Die Attribute, die allen gefundenen Produkten gemeinsam sind, werden mit der Relevanz jeweils als Spaltenüberschrift der Tabelle repräsentiert, während die Ausprägungen von diesen Attributen eines Produktes als Tabelleneinträge in einer Tabellenzeile repräsentiert werden. Die weiteren Attribute und deren Ausprägungen vom jeweiligen Produkt werden von PIA als Text zusammengefaßt und in einer weiteren Tabellenzeile dargestellt (vgl. Abb. 2.10).

Abbildung 2.10: Suchergebnis anzeigen

2.3.1.2.3 Ergebnis formatieren

Wenn der Benutzer das Ergebnis auf andere Weise darstellen will, kann er mit Hilfe einer Registerkarte, die in Abbildung 2.11 illustriert wird, folgende Einstellungen festlegen bzw. ändern:

- welche Attribute in Spalten angezeigt werden sollen,
- wie breit die jeweilige Spalte sein soll,

- welche Treffer überhaupt angezeigt werden sollen, indem er eine reelle Zahl zwischen 0% und 100% angibt, so daß nur die Produkte angezeigt werden sollen, deren Relevanz darüber liegt,
- in welcher Reihenfolge die Treffer sortiert werden sollen, z.B. nach der Ausprägung eines bestimmten Attributes,
- mit welcher Schriftart, -größe und -farbe die Treffer dargestellt werden sollen sowie
- ob die Treffer nach einem bestimmten Attribut gruppiert werden sollen.

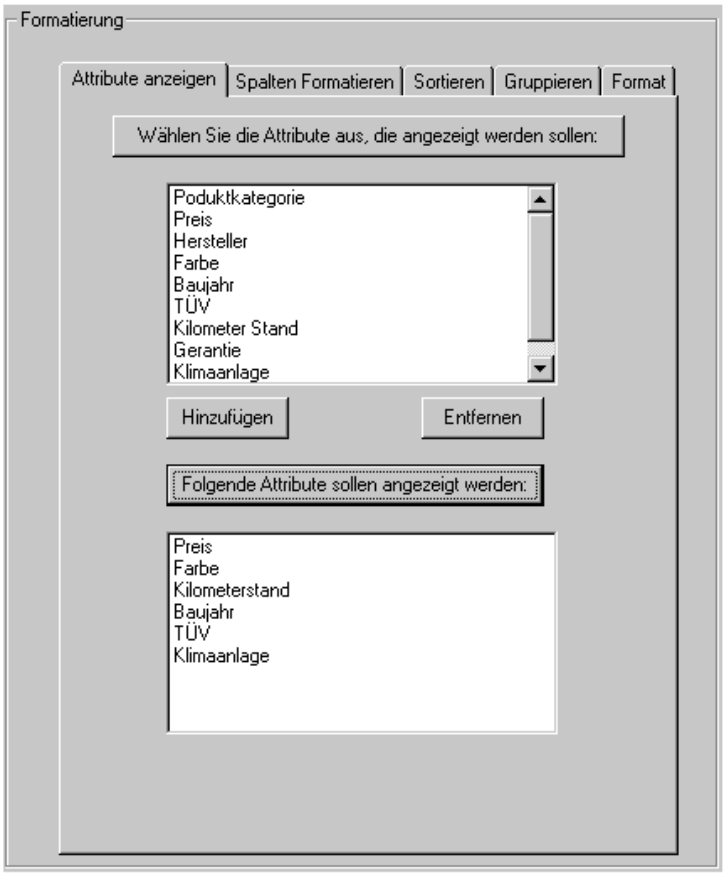

Abbildung 2.11: Ergebnis Formatierung

2.3.1.2.4 Suchaufträge speichern/abonnieren

Der Benutzer kann seinen Suchauftrag bei PIA speichern lassen, falls er sich weiterhin für solche Produkte interessiert. Wenn sich der Benutzer beim nächsten Mal am PIA anmeldet, sieht er sofort alle seine in PIA gespeicherten Suchaufträge, die er direkt noch mal auswerten lassen kann.

Da die Datenbestände in der Produktdatenbank ständig aktualisiert werden, sei es durch Einfügen von neuen Produkten oder durch Verändern gewisser Attribute von vorhandenen Produkten, bietet der PIA Benutzern die Möglichkeit an, Suchaufträge zu abonnieren. Ein Abonnement bedeutet in diesem Fall, daß der Suchauftrag in der Datenbank gespeichert und in regelmäßigen Zeitabständen ausgeführt wird. Dadurch kann ein Benutzer ständig über die

Änderungen an den von ihm angefragten Produkten informiert werden, ohne sich immer wieder am System anmelden und manuell Anfragen stellen zu müssen.

Beim Abonnieren eines Suchauftrages vereinbart der Benutzer mit PIA:

- in welchem Umfang,
- auf welche Weise,
- wie oft ihm das Suchergebnis geschickt werden soll und
- für wie lange das Abonnement gelten soll.

2.3.1.3 Anwendungsfall Bearbeitung Suchaufträge

Ist ein Benutzer dem System bekannt, kann er nach der Anmeldung seine sämtlichen gespeicherten und abonnierten Suchaufträge anzeigen lassen. Zu jedem Suchauftrag gibt es drei graphische Steuerungselemente, mit denen der Benutzer jeweils das Suchergebnis anzeigen lassen („Anzeigen"), den Suchauftrag ändern („Ändern") oder löschen („Löschen") kann.

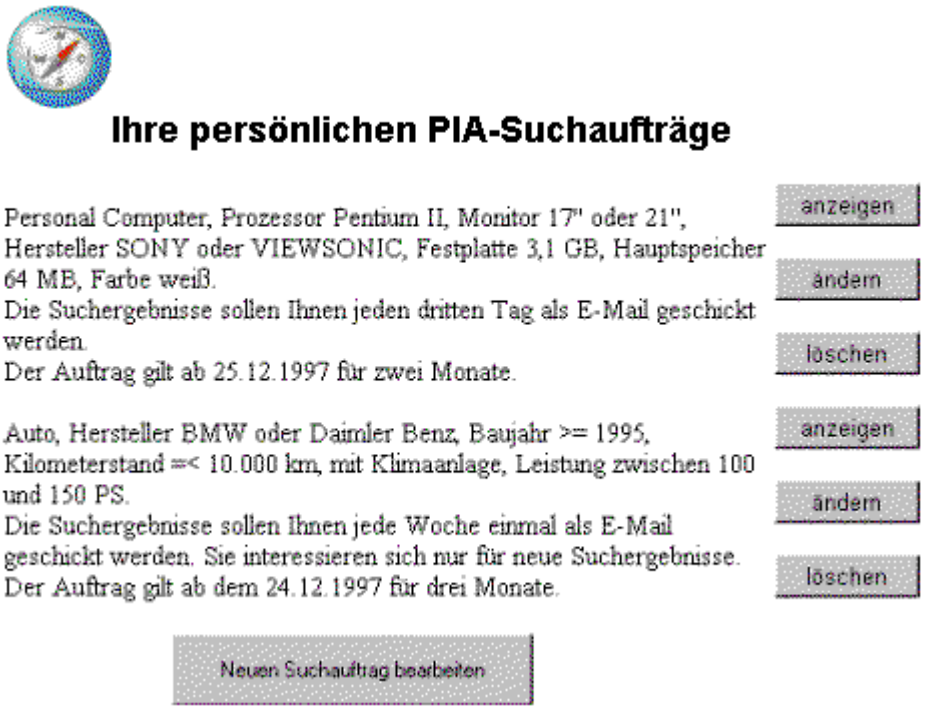

Abbildung 2.12: Benutzerschnittstelle für den Anwendungsfall Bearbeitung Suchaufträge

Ein Benutzer kann einen Suchauftrag ändern, indem er auf „Ändern" neben dem jeweiligen Suchauftrag klickt. Zum Ändern eines Suchauftrages gehört:

- die Vereinbarungen mit PIA beim Abonnement zu ändern (z.B. Rhythmus, Umfang, Zeitraum etc.),
- das Ergebnisformat anders einzustellen sowie
- bestimmte Attributanfragen in dem Suchauftrag zu ändern (z.B. Löschen einer Attributanfrage aus dem Suchauftrag, Änderung der Ausprägung eines Attributs, Hinzufügen neuer Attributanfrage zu dem Suchauftrag usw.).

Die gespeicherten Suchaufträge sowie die Abonnements kann er ebenfalls jederzeit löschen, indem er einfach auf den „Löschen" Button hinter dem zu löschenden Suchauftrag klickt.

Der gesamte Ablauf vom Anwendungsfall „Bearbeitung eines Suchauftrages“ wird durch das folgende Diagramm beschrieben:

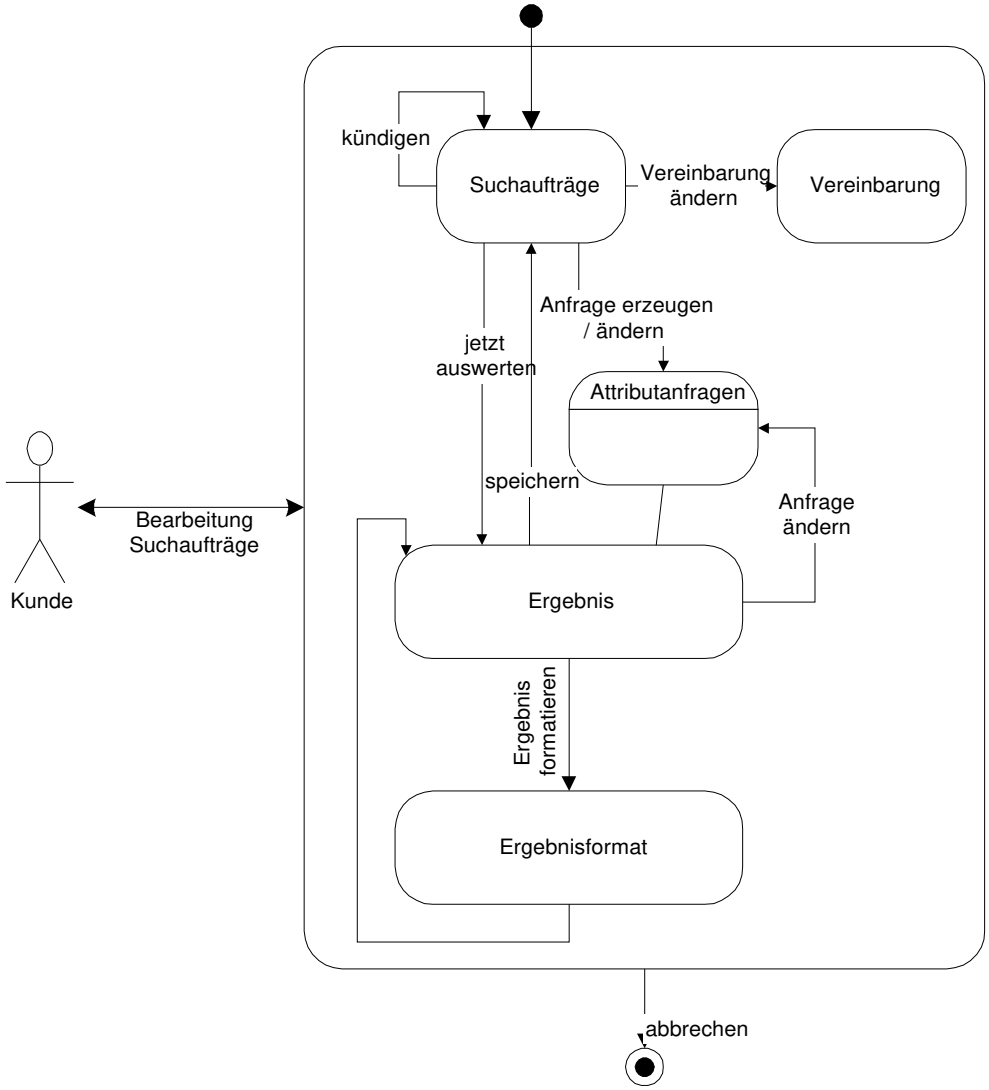

Abbildung 2.13: Zustandsdiagramm Bearbeitung Suchaufträge

2.3.2 Weitere Anwendungsfälle für Anbieter

In den vorigen Abschnitten werden die gemeinsamen Anwendungsfälle für „Kunde“ und „Anbieter“ beschrieben. Außer Suchaufträge zu erstellen, kann ein Anbieter seine Produkte in PIA einfügen, die von ihm eingefügten Produkte löschen oder deren Attribute/Ausprägungen verändern. Diese drei Aktivitäten werden unter dem Begriff “Produktpflege“ zusammengefaßt. Dementsprechend ist jedem Anbieter in PIA eine Produktliste zugeordnet, in der die Daten aller von ihm eingefügten Produkte gespeichert werden. Nachdem sich ein Anbieter am PIA angemeldet hat, kann er entweder:

- Suchaufträge erstellen, bearbeiten oder
- seine eigene Produktliste bearbeiten (vgl. Abb. 2.14).

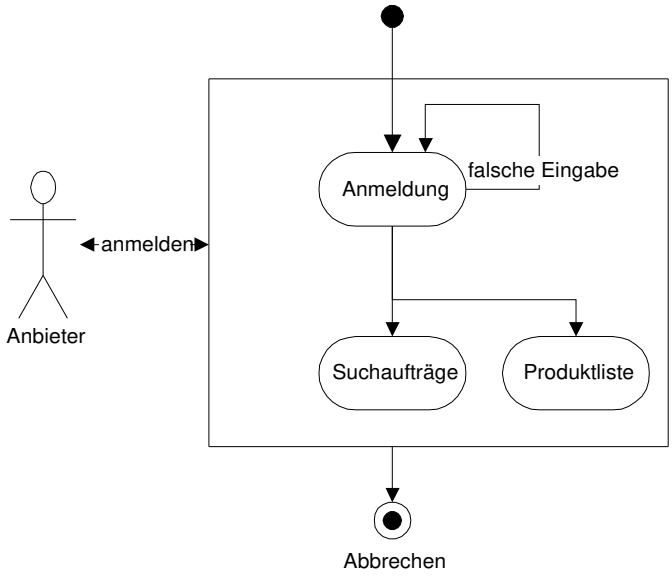

Abbildung 2.14: Zustandsdiagramm des Anwendungsfalls Anbieteranmeldung

2.3.2.1 Produkte einfügen

Ein Produkt wird in PIA durch Attribute und deren Ausprägungen bestimmt. Ein Beispiel dafür ist:

Siemens Nixdorf 888:

Attribut	**Ausprägung**
Produktkategorie	Personal Computer
Preis	4.500,00 DM
Hauptspeicher	64 MB
Festplatte	3,2 GB
Prozessor	Intel Pentium 166 MHZ
Betriebssystem	Windows 95

Beim Einfügen eines Produktes sucht der Anbieter die Attribute aus, die seiner Meinung nach für sein Produkt relevant sind, und gibt die Ausprägungen für diese Attribute an. Der Unterschied zwischen „Produkt einfügen“ und „Produkt suchen“ ist, daß der Anbieter nicht nur auf die schon im System vorhandenen Attribute zurückgreifen kann, sondern auch eigene, neue Attribute verwenden darf, die seiner Meinung nach die jeweiligen Produkteigenschaften seines Produkts am besten repräsentieren. Neue Attribute werden vom Redakteur in PIA eingefügt[3]. Zu beachten ist, daß der Anbieter für ein Attribut nur einen präzisen Wert angeben kann – eine Attributausprägung, während er beim Suchen nach Produkten eine Menge von Ausprägungen angeben kann.

2.3.2.2 Produkte ändern/löschen

Ein Anbieter kann jederzeit die Daten der von ihm eingefügten Produkte ändern. Jedem Produkt ist sein Anbieter zugeordnet. Nach der Anmeldung eines Anbieters kann dieser seine Suchaufträge bearbeiten oder seine Produktangebote pflegen. Für den ersten Fall agiert er wie

[3] Vgl. Abschnitt 2.3.3: Anwendungsfälle für Redakteur

ein normaler Kunde und bekommt eine Auflistung von seinen Suchaufträgen[4]. Für den zweiten Fall werden ihm die Produkte, die er in PIA eingefügt hat, aufgelistet.

Die Benutzerschnittstelle für Produktpflege ist ähnlich wie beim „Produkt suchen“. Zu jedem Produkt, das der Anbieter eingefügt hat, gibt es eine Gruppe von Steuerungselementen:

- Ändern und
- Löschen.

Wenn der Anbieter auf „Ändern“ klickt, wird das ausgewählte Produkt in einer Tabelle auf einer einzigen Seite dargestellt, wobei jede Tabellenzeile eine Attributausprägung repräsentiert - in der ersten Spalte das Attribut, in der zweiten Spalte die Ausprägung. Der Anbieter kann ein Attribut umbenennen oder eine Ausprägung eines Attributes ändern, indem er die alte Eintragung in der entsprechenden Tabellenzelle überschreibt. Außerdem kann der Anbieter neue Zeilen in die Tabelle einfügen, wenn er das Produkt durch weitere Attribute beschreiben möchte.

Durch Klicken auf den „Löschen“ Button neben einem Produkt wird das Produkt gelöscht.

Folgendes Diagramm illustriert den Ablauf der Anwendungsfälle Produktpflege.

[4] Vgl. Abschnitt 2.3.4.3.

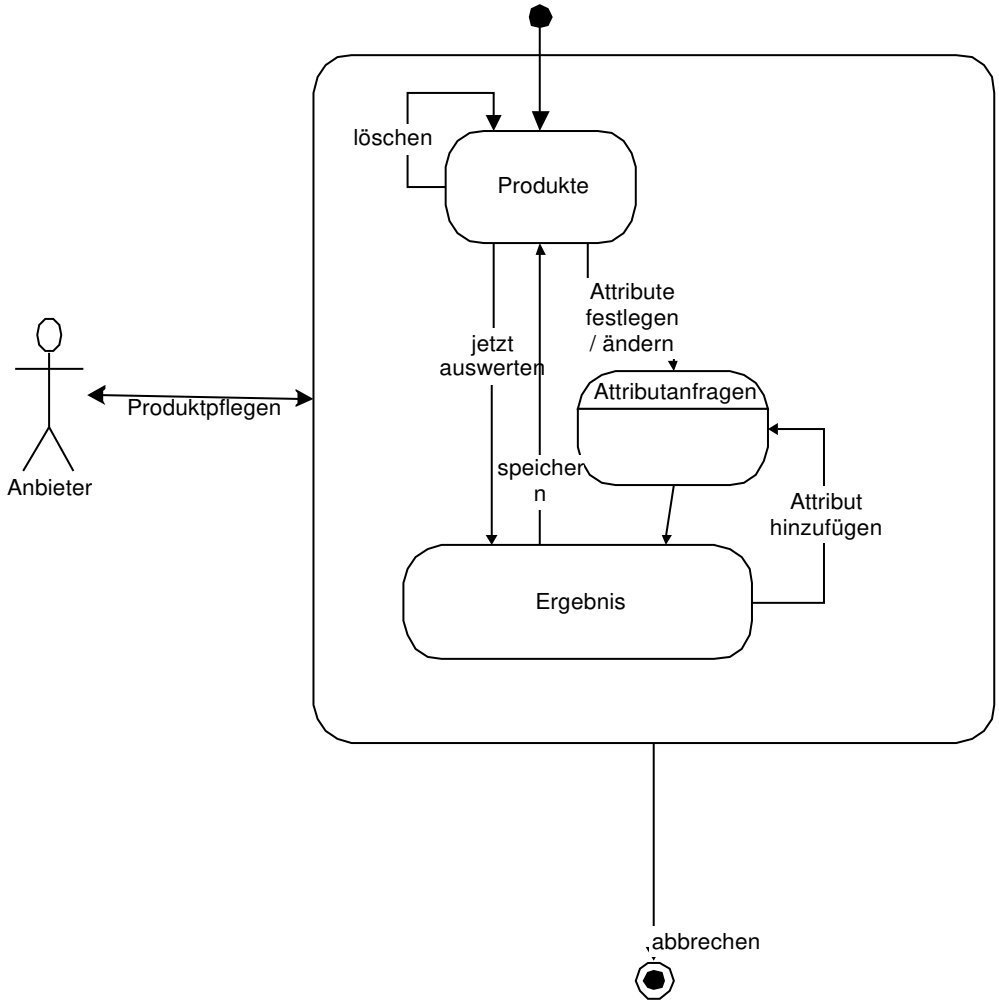

Abbildung 2.15: Zustandsdiagramm Produktpflege

2.3.3 Anwendungsfälle für Redakteur

Der Redakteur ist für die Verwaltung von Attributen/Domänen zuständig.

2.3.3.1 Domänen

Eine Domäne beschreibt die Menge aller möglichen Ausprägungen für ein Attribut.

Wir unterscheiden in PIA zwischen den folgenden vier Domänen:

- Skalar,
- hierarchische Domäne,
- Aufzählungsdomäne und
- Volltextdomäne.

2.3.3.1.1 Skalar

Ein Skalar ist eine reelle Domäne, die gleichartige technische, physikalische, chemische, ... Größen beschreibt (Beispiel: Temperatur).

Eine Ausprägung eines skalaren Attributs wird durch ein Verhältnis der jeweiligen Größe zu einer vereinbarten Maßeinheit beschrieben (Beispiel: 34 Grad Celsius)

Für ein Skalar können mehrere Einheiten definiert sein (Beispiel: Grad Celsius und Grad Fahrenheit). Davon ist eine Einheit als "Referenzeinheit" ausgezeichnet. Alle anderen Einheiten E werden als abgeleitete Einheiten bezeichnet und müssen sich durch eine Formel der Form

1 Referenzeinheit = Faktor * E + Summand

darstellen lassen. Beispiel:

1 Grad Celsius = 9/5 * Fahrenheit + 32

2.3.3.1.2 Hierarchische Domäne

Eine hierarchische Domäne läßt sich in eine Hierarchie von Domänen einordnen (z.B. Land).

Eine hierarchische Domäne hat Unterdomänen, die diese Domäne spezialisieren (z.B. europäische Länder, asiatische Länder usw.).

2.3.3.1.3 Aufzählungsdomäne

Eine Aufzählungsdomäne ist ein Aufzählungstyp.

Eine Aufzählungsdomäne ist eine aufzählbare Menge von Attributausprägungen (z.B. Prozessortyp, davon gibt es auf dem Markt nur aufzählbare Varianten.).

2.3.3.1.4 Volltextdomäne

Eine Volltextdomäne besteht aus Volltextwerten.

2.3.3.2 Verwaltung Attribute/Domäne

Immer, wenn ein neues Attribut in PIA eingefügt wird, muß gleichzeitig auch die zulässige Wertmenge für dieses Attribut festgelegt werden. Verwaltung von Attributen und Domänen sind zwei Aufgaben, die sich nicht voneinander trennen lassen.

Unter „Verwalten der Attribute/Domänen“ versteht man:

- Attribute/Domänen einfügen
- Attribute/Domänen ändern
- Attribute/Domänen löschen.

2.3.3.2.1 Einfügen Attribute/Domänen

Anhand der Eingabedaten der Anbieter entscheidet der Redakteur, ob es notwendig ist, neue Attribute (Domänen) hinzuzufügen. Meistens ist es der Fall, wenn ein neues Attribut eingefügt wird, existiert noch keine passende Domäne für dieses Attribut im System. D.h., daß der Redakteur beim Einfügen eines neuen Attributes gleichzeitig eine neue Domäne für dieses Attribut definiert. Der Redakteur entscheidet, was für eine Art Attribut er einfügt.

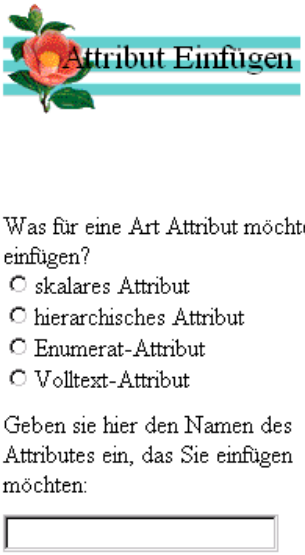

Abbildung 2.16: Einfügen eines Attributes

2.3.3.2.1.1 Einfügen skalarer Attribute/Domänen

Beim Einfügen eines skalaren Attributes gibt der Redakteur zuerst einen Namen für dieses Attribut an (z.B. Preis). Dann definiert er noch eine neue Domäne für dieses Attribut, indem er zulässige Werte für die Ausprägungen festlegt (z.B. positive reelle Zahlen). Weiterhin muß er die Menge der zulässigen Einheiten für diese skalaren Werte angeben (z.B. Dollar, DM usw.), darunter eine spezielle Einheit als Referenzeinheit. Für die anderen Einheiten werden dann die Umrechnungsformeln (Faktor und Summand, vgl. Abschnitt 2.1.4.1 Sklare Domäne) auch gleichzeitig definiert.

2.3.3.2.1.2 Einfügen hierarchischer Attribute/Domäne

Beim Einfügen eines hierarchischen Attributs gibt der Redakteur ebenfalls den Namen für dieses Attribut an (z.B. Land). Dann fügt er noch eine hierarchische Domäne in das System ein, indem er die übergeordnete Domäne festlegt, wie z.B. im folgenden Diagramm:

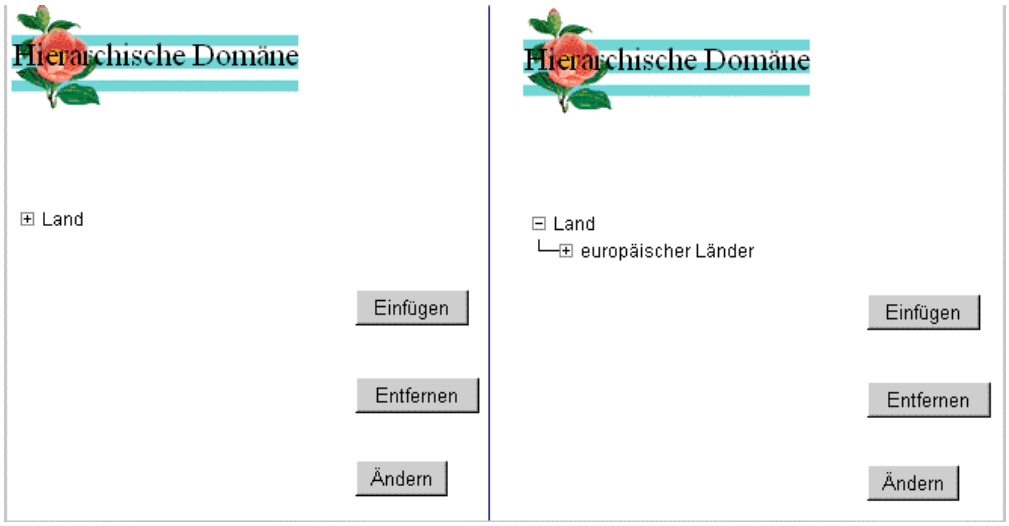

Abbildung 2.17: Einfügen Hierarchischer Domäne

Wenn der Redakteur „Land“ markiert und dann auf „Einfügen“ klickt, kann er die Menge der Ausprägungen für die hierarchische Domäne „Land“ erweitern (z.B. europäische Länder).

2.3.3.2.1.3 Einfügen von Aufzählungsattribute/-domänen

Der Redakteur gibt den Namen des Attributs an (z.B. Prozessor), dann definiert er eine dazugehörige Domäne (z.B. Prozessor), indem er alle möglichen Ausprägungen aufzählt.

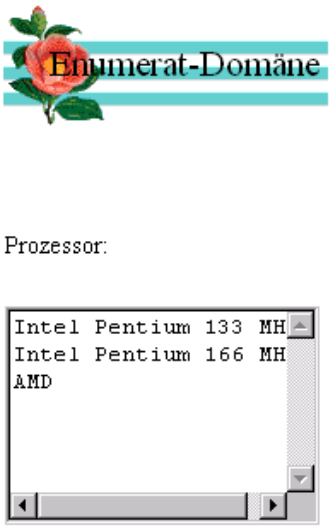

Abbildung 2.18: Einfügen Aufzählungs-Domäne

2.3.3.2.1.4 Einfügen eines Volltextattributes

Hier muß der Redakteur nur den Namen des einzufügenden Attribut einzugeben.

2.3.3.2.2 Löschen Attribute/Domänen

Nach dem Anmelden wird dem Redakteur die Liste der Attribute im System angezeigt. Beim Löschen eines Attribut braucht er nur das Attribut zu markieren und dann auf den „Löschen“ Button zu klicken.

2.3.3.2.3 Ändern Attribute/Domänen

Wenn der Redakteur ein Attribut markiert und dann auf den „Ändern“ Button klickt, stell ihm das System jeweils eine „domänenspezifische“ Benutzerschnittstelle dar, die im folgenden beschrieben wird.

2.3.3.2.3.1 Ändern eines skalares Attribut/einer skalarer Domäne

Dazu gehören:

- Referenzeinheit ändern, PIA ändert dann automatisch die Umrechnungsfaktoren,
- eine oder mehrere Einheiten entfernen/hinzufügen,
- die Faktoren und Summanden für manche Einheiten aktualisieren (z.B. Währung),
- ein Attribut/eine Domäne umbenennen,
- die Wertmenge verkleinern/vergrößern.

2.3.3.2.3.2 Ändern eines hierarchisches Attributes/einer hierarchischen Domäne

Beim Ändern hierarchischer Attribute/Domänen treten mehrere Fälle auf, z.B.:

- Verschieben einer Unterdomäne zu einer anderen Superdomäne,
- Kopie einer Domäne zu einer anderen Superdomäne,
- Umbenennen einer Domäne sowie
- die Wertmenge verkleinern/vergrößern.

2.3.3.2.3.3 Ändern eines Aufzählungsattributs/einer Aufzählungsdomäne

Hierzu gehören:

- eine oder mehrere Aufzählung(-en) von/zu einer Aufzählungsdomäne hinzufügen/entfernen oder
- die Domäne umbenennen.

2.3.3.2.3.4 Ändern eines Volltextattributs/einer Volltextdomäne

Beim Ändern eines Volltextattributs bzw. einer Volltextdomäne kann der Redakteur das Attribut bzw. die Domäne umbenennen.

2.3.4 Anwendungsfälle für Administrator

Der Administrator ist für die Benutzerverwaltung zuständig. Er kann

- neue Benutzer einfügen,
- Benutzer löschen oder

- Berechtigungen an Benutzer vergeben.

Nach der Anmeldung wird dem Administrator die gesamten Benutzer aufgelistet. Hinter jedem Benutzer stehen zwei Steuerungselemente – „Ändern“ und „Löschen“, mit deren Hilfe der Administrator die Daten eines Benutzers ändern oder einen Benutzer löschen kann.

2.3.4.1 Neue Benutzer einfügen

Die vom Benutzer beim Erstellen eines Benutzerantrages eingegebenen Daten werden an dem Administrator weiter geleitet. Er prüft diese und gibt sie in den PIA ein. Dann wird ein neues Account für den Benutzer eingerichtet. Dabei wird der neue Benutzer einer oder mehreren der 4 Gruppen (Kunde, Anbieter, Redakteur, Administrator) zugeordnet. Der Benutzer kann auch jederzeit einen neuen Antrag stellen, wenn er seine Benutzergruppe noch ändern möchte, z.B. wenn ein Kunde Produkte anbieten möchte.

Wenn sich ein Benutzer in das System einloggt und nach Produkten sucht oder Produktangebote macht oder das System verläßt, wird dieses vom System protokolliert. Der Administrator archiviert die Daten, an welchem Tag zu welcher Zeit welcher Benutzer nach welchen Produkten gesucht hat/welche Produktangebote gemacht hat usw. Wenn ein Benutzer für längere Zeit seinen Account nicht benutzt hat, bekommt er eine E-Mail vom Administrator, der ihn um eine Rückmeldung bittet. Wenn der Administrator aber innerhalb eines bestimmten Zeitraums keine Antwort erhält, wird der Account gelöscht.

2.3.4.2 Benutzer ändern/löschen

Nach dem Anmelden am PIA werden dem Administrator die Benutzernamen aufgelistet, er kann einen davon aussuchen und diesen bearbeiten – Benutzer ändern oder löschen.

Zum Ändern von Benutzer gehören:

- Benutzergruppe ändern und
- Benutzerdaten ändern (z.B. Beruf, Adresse etc.).

Beim Löschen von Benutzers wählt der Administrator den Benutzernamen des zu löschenden Benutzers aus und klickt auf den „Löschen“ Button, der wie „Ändern“ als Steuerungselement hinter dem Benutzer steht.

2.4 Zusammenfassung der Analysephase

In diesem Kapitel wird ein Anwendungsfallmodell entwickelt, das alle Akteure des PIA Systems, sämtliche Anwendungsfälle sowie die Beziehungen zwischen den Akteuren und den Anwendungsfällen beschreibt. Beim Aufbau dieses Modells fängt man mit der Identifikation der Akteure an, die kein Bestandteil des zu entwickelnden Systems sind und nur diejenigen repräsentieren, die mit dem System kommunizieren. Sie werden später als Hilfsmittel für die Identifikation der Anwendungsfälle benutzt. Die Anwendungsfälle repräsentieren hingegen die Funktionalitäten des Systems und modellieren einen Dialog zwischen einem Akteur und dem System.

Kapitel 3

Objektmodellierung

Die objektorientierte Technologie stellt eine praxisnahe und produktive Möglichkeit dar, Software für die meisten Anwendungen unabhängig von der Implementierungssprache zu entwickeln [Rumbaugh 93]. Der objektorientierte Ansatz basiert darauf, Objekte der realen Welt durch Daten, Operationen bzw. Umgangsformen zu modellieren, und danach das entstandene Modell zur Entwicklung zu verwenden.. Objektorientierte Modellierung und objektorientierter Entwurf sind eine neue Art der Problemlösung, in deren Mittelpunkt Modelle stehen, die Konzepte der realen Welt übernehmen.

Im zweiten Kapitel wird ein Analysemodell entwickelt, in dem die Anforderungen an den PIA beschrieben worden sind, das aber nicht die Umsetzung beinhaltet. In diesem Kapitel werden beim Übergang vom Analysemodell zum Objektmodell die aus dem Use-Case-Modell herauskristallisierten Objektklassen in einem ersten Klassendiagramm fixiert und ihre Beziehungen untereinander beschrieben.

Ein Objektmodell erfaßt die statische Struktur eines Systems. Es zeigt die Objekte im System, die Relationen zwischen den Objekten mittels Assoziationen und die Attribute und Operationen, die jede Klasse von Objekten charakterisieren. Ein Objekt wird definiert als Konzept, Abstraktion oder Gegenstand mit klaren Abgrenzungen und einer präzisen Bedeutung für das anstehende Problem. Eine Klasse beschreibt hingegen ein Gruppe von Objekten mit ähnlichen Eigenschaften (Attributen), gemeinsamem Verhalten (Operationen), gemeinsamen Relationen zu anderen Objekten und einer gemeinsamen Semantik. Bei Assoziationen handelt es sich um Vererbungs-, Aggregat- und Benutztbeziehungen zwischen den Klassen. Jede Assoziation besitzt zusätzlich eine Kardinalität, die angibt, ob es sich um eine 1:1, 1:n oder n:m-Beziehung handelt.

Es werden drei Objekttypen verwendet: Entity, Interface und Control, um Objektklassen zu identifizieren und diesen Klassen Funktionen zuzuweisen, damit ein stabiles System entsteht und Veränderungen soweit wie möglich auf lokaler Ebene stattfinden können [Yourdan 96].

- Entity-Objekte entsprechen wirklichen Objekten oder Objekten, die man direkt im Geschäftsbereich erkennen kann.

- Interface-Objekte werden vom System zur Kommunikation mit externen Komponenten verwendet, um Funktionen einzukapseln, die von der Umgebung abhängig sind. Sie isolieren Funktionen, die zu Anwender- und Maschinenschnittstellen gehören, so daß nur die Interface- und nicht die Entity-Objekte von Detailänderungen der Schnittstellen betroffen sind[5].

- Control-Objekte enthalten Operationen, die normalerweise nicht bei Entity- oder Interface-Objekten zu finden sind. Solche Funktionen beziehen sich auf Ereignisse, die eine Reihe von Objekten betreffen und bei denen die Operationen nicht einem einzigen Entity-Objekt zugeordnet sind.

[5] Vgl. Abschnitt 3.2: Java Interface Klassen

3.1 Die Objekt-Klassen

Das Hauptanliegen bei der objektorientierten Modellierung in dieser Phase ist die Erstellung eines vollständigen Modells für die Entity-Objekte. Im folgenden werden die Arbeitsschritte für den Einbau konkreter Objekte in das Modell beschrieben:

- Objektklassen identifizieren,
- Assoziationen identifizieren,
- Attribute für Objekte identifizieren,
- Benennung und Spezifikation der Operationen,
- Verfeinerung der Beziehungen durch Vererbung und Aggregation

Die folgende Abbildung stellt einen Überblick über die in PIA vorhandenen Entity-Klassen in OMT dar. Diese werden in den folgenden Abschnitten weiter geteilt und detailliert beschrieben.

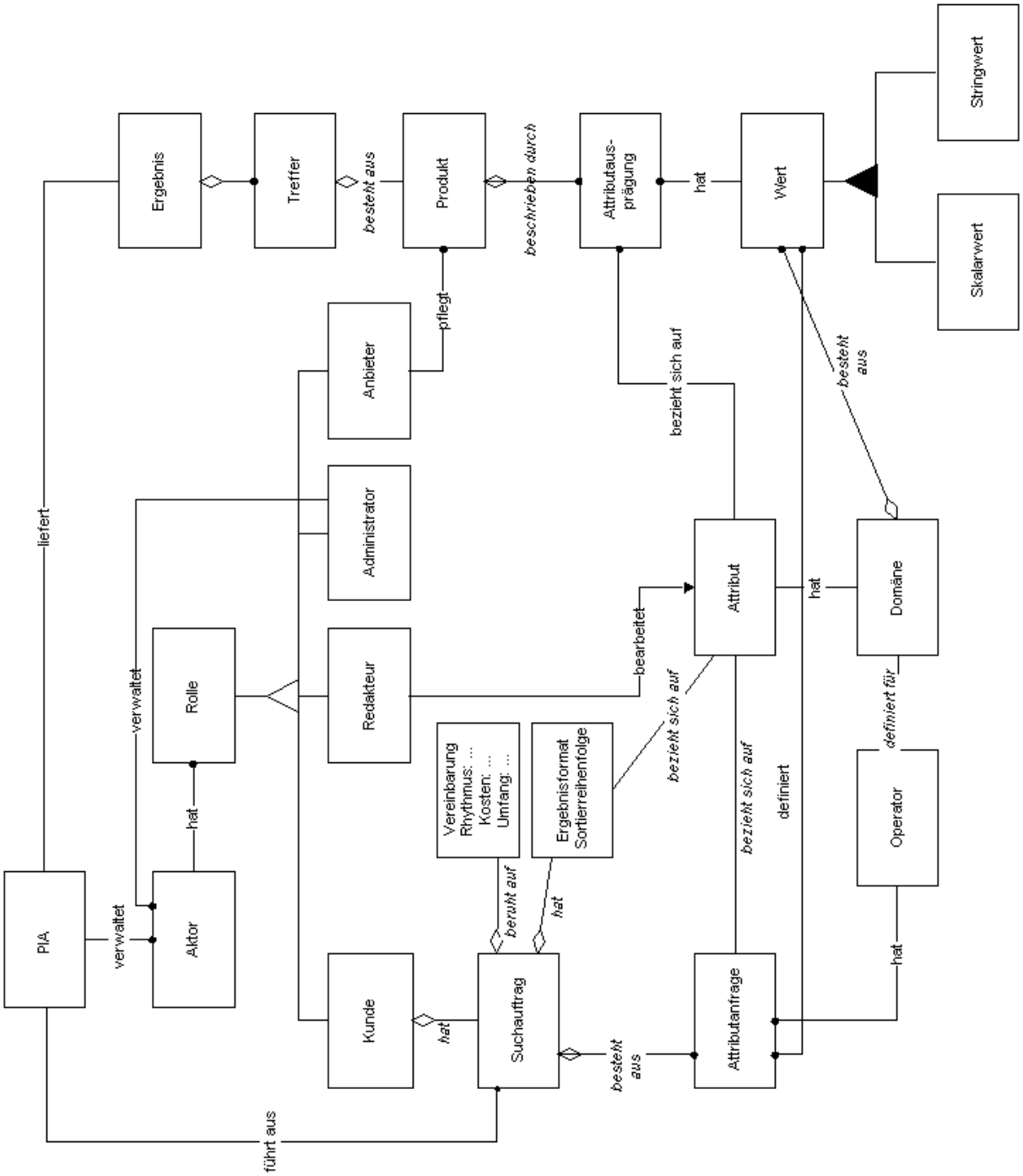

Abbildung 3.1: PIA Klassen in Überblick

3.1.1 Die Klassen Akteur, Rolle

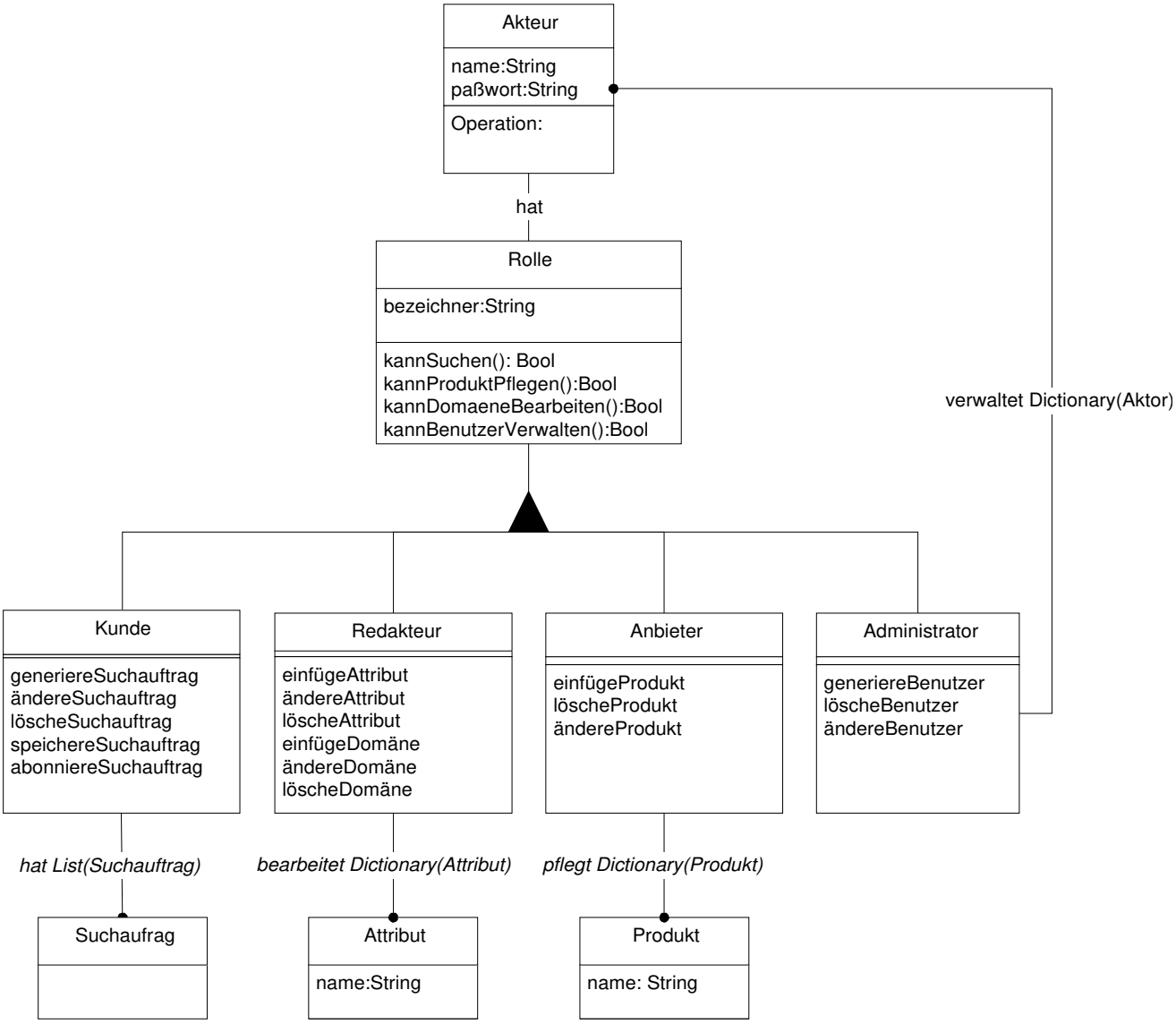

Abbildung 3.2: Die Klassen Akteur, Rolle

Ein Akteursobjekt bildet einen Benutzer der realen Welt ab und wird durch einen Benutzernamen und ein Paßwort identifiziert. Ein Akteur kann eine oder mehrere Rollen besitzen.

Ein großer Bestandteil der Objektorientierung ist die Polymorphie. Durch Polymorphie sind alle Objekte einer Unterklasse typkompatibel mit Objekten der Superklasse. Daher können sie immer da verwendet werden, wo Objekte der Superklasse erwartet werden. Da sie aber die Operationen der Superklasse neu definieren können, zeigen sie meistens auch ein anderes Verhalten als Objekte der Superklasse. Bei der Programmierung braucht sich der Programmierer nicht darum zu kümmern, mit Objekten welcher konkreten Unterklassen er zu tun hat, weil die Bindung an die konkrete Implementierung dynamisch zur Laufzeit geschieht. Da wir nicht überall die Fallunterscheidung über die vier verschiedene Rollentypen machen wollen, wird eine abstrakte Klasse Rolle definiert, die in erster Linie nur das gemeinsame

Verhalten aller Unterklassen definiert, aber nicht implementiert. Dadurch wird erreicht, daß alle Unterklassen von Rolle ein gemeinsames Verhalten zeigen und daher im System einheitlich behandelt werden können.

Die Methoden *kannSuchen()*, *kannProduktPflegen()*, *kannDomänenBearbeiten()* sowie *kannBenutzerVerwalten()* der Klasse Rolle werden hier nur definiert. Alle Unterklassen von der Klasse Rolle müssen diese Methoden implementieren, damit sie typkompatibel mit Rolle sind. Die Klasse Rolle wird damit als eine abstrakte Klasse bezeichnet, die selbst keine direkte Instanzen hat, deren Nachkommen aber direkte Instanzen besitzen.

Ein Akteursobjekt mit der Rolle „Kunde“ darf PIA Suchaufträge erstellen, Suchaufträge speichern lassen sowie sie abonnieren. Dagegen kann ein Akteursobjekt mit der Rolle „Anbieter“ Produkte in PIA einfügen und die Daten der von ihm eingefügten Produkte bearbeiten. Wenn ein Akteur die Rolle „Redakteur“ besitzt, kann er Attribute und Domänen bearbeiten. Und ein Akteursobjekt mit der Rolle „Administrator“ darf neue Benutzer in PIA einfügen sowie die alten Benutzer löschen oder die Berechtigungen vergeben.

Die Suchaufträge eines Kunden bzw. die Produkte eines Anbieters werden intern als eine Liste oder ein Dictionary in dem Rollenobjekt gespeichert. Damit wird es gewährleistet, daß die Anbieter oder Kunden nicht auf Produkte oder Suchaufträge, die von anderen Anbietern und Kunden stammen, zugreifen können. Die Klasse Liste und die Klasse Dictionary unterscheiden sich darin, daß ein Element einer Dictionary einen eindeutigen Schlüssel hat, während es bei einem Element der Liste nicht der Fall ist.

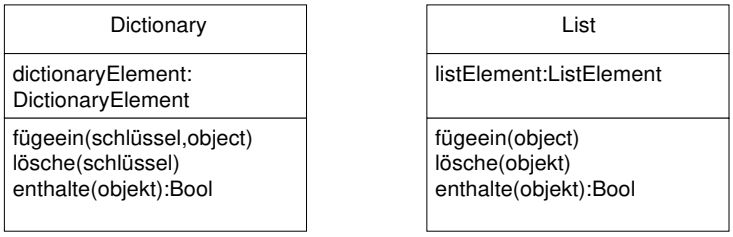

Abbildung 3.3: Die Klasse Dictionary und die Klasse List

3.1.2 Die Klassen Kunde, Suchauftrag

Die Methoden der Klasse Kunde sind:

- *generiereSuchauftrag(): Suchauftrag*

Durch diese Methode wird ein leerer Suchauftrag erzeugt. Die Methode wird aufgerufen, wenn der Benutzer auf „neuen Suchauftrag bearbeiten“ klickt.

- *löscheSuchauftrag(s: Suchauftrag)*

  Die nicht mehr benötigten Aufträge werden durch Aufruf dieser Methode aus der Liste der Suchaufträge entfernt.

- *ändereSuchauftrag(s: Suchauftrag)*

Mit dieser Methode kann der als Parameter mitgegebene Suchauftrag modifiziert werden.

- *speichereSuchauftrag (s: Suchauftrag)*

  Diese Methode speichert den Suchauftrag in die Liste der Suchaufträge von dem Kundeobjekt ein.

- *abonniereSuchauftrag (s: Suchauftrag)*

  Damit wird der Suchauftrag in der Abonnementliste des Kunden eingetragen.

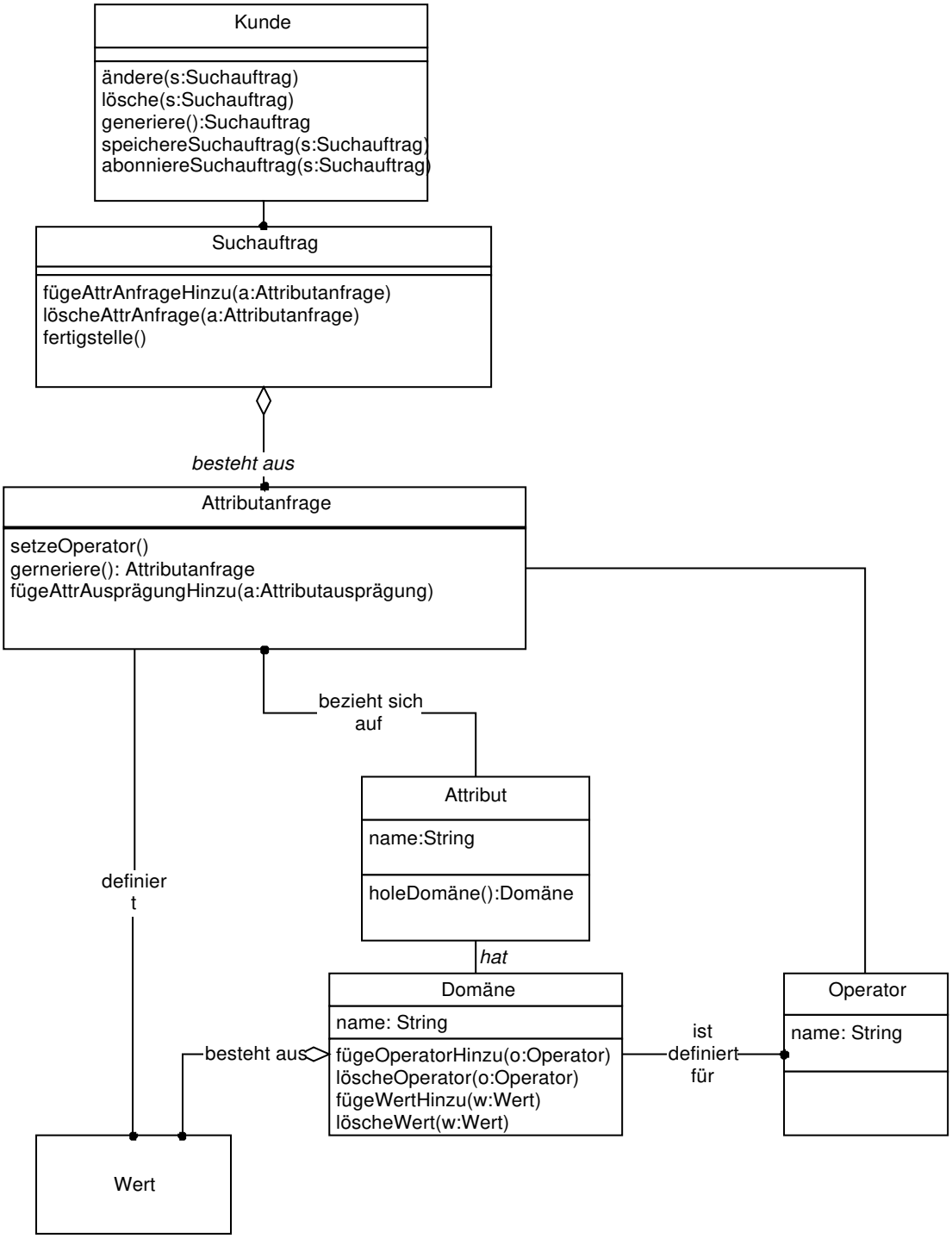

Abbildung 3.4: Die Klassen Kunde, Suchauftrag, Attributanfrage

3.1.3 Die Klasse Suchauftrag

Ein Suchauftrag ist ein Behälterobjekt und besteht aus mehreren Attributanfragen, wobei eine Attributanfrage aus einem Attribut, einem Operator und einem einzigen Attributwert oder einer Liste von Attributwerten besteht. Einem Objekt der Klasse Suchauftrag sind folgende Methoden zugeordnet:

- *fügeAttrAnfrageHinzu(a:Attributanfrage)*

  Sobald ein Attribut und ein Operator ausgesucht werden und die dazugehörigen Ausprägungen festgelegt wird, wird ein Objekt der Klasse Attributanfrage erzeugt und die Methode *fügeAttrAnfrageHinzu(a: Attributanfrage)* des Suchauftragobjekts wird aufgerufen, in der das Objekt Suchauftrag diese Attributanfrage in seine interne Liste von Attributanfragen aufnimmt.

- *löscheAttrAnfrage(a:Attributanfrage)*

  Wenn der Benutzer ein Attribut, das er ausgesucht hat, und die dazugehörigen Ausprägungen wieder aus dem Suchauftrag entfernen will, wird die Methode des Suchauftrags löscheAttrAnfrage(a: Attributanfrage) aufgerufen, und damit dieses Attributanfrageobjekt aus dem Suchauftragobjekt herausgenommen.

- *stellefertig ()*

  Diese Methode wird aufgerufen, wenn der Benutzer das Suchergebnis auswerten lassen möchte. Dadurch wird ein Suchauftragobjekt erzeugt, das von PIA ausgeführt wird.

3.1.4 Die Klasse Attributanfrage

Ein Objekt der Klasse Attributanfrage bezieht sich auf ein Objekt der Klasse Attribut. Eine Attributanfrage besteht aus einem Attribut und einem Operator, der zu diesem Attribut paßt. Die zu einem Attribut passenden Operatoren werden in dem Domänenobjekt, das zu dem Attribut gehört, gespeichert. Eine Attributanfrage enthält weitere Wertangaben zu dem Attribut, nach denen die Produkte gesucht werden sollen.

Für eine skalare Domäne sind im PIA folgende Operatoren definiert:

- >= (in Worten „mindestens gleich“),
- ~= (in Worten „ungefähr gleich“),
- >= ... =<... (in Worten „zwischen ... und ...“) und
- =< (in Worten „höchstens gleich“).

Für eine hierarchische Domäne oder eine Aufzählungsdomäne ist der Operator “∈“ definiert, was die Beziehung „ist Element von“ beschreibt.

Für eine Volltextdomäne ist der Operator „enthalten“ definiert. Ein Beispiel ist, daß eine Produktbeschreibung das Wort „Otto“ enthalten soll.

3.1.5 Das Klassendiagramm von Redakteur, Attribut, Domäne

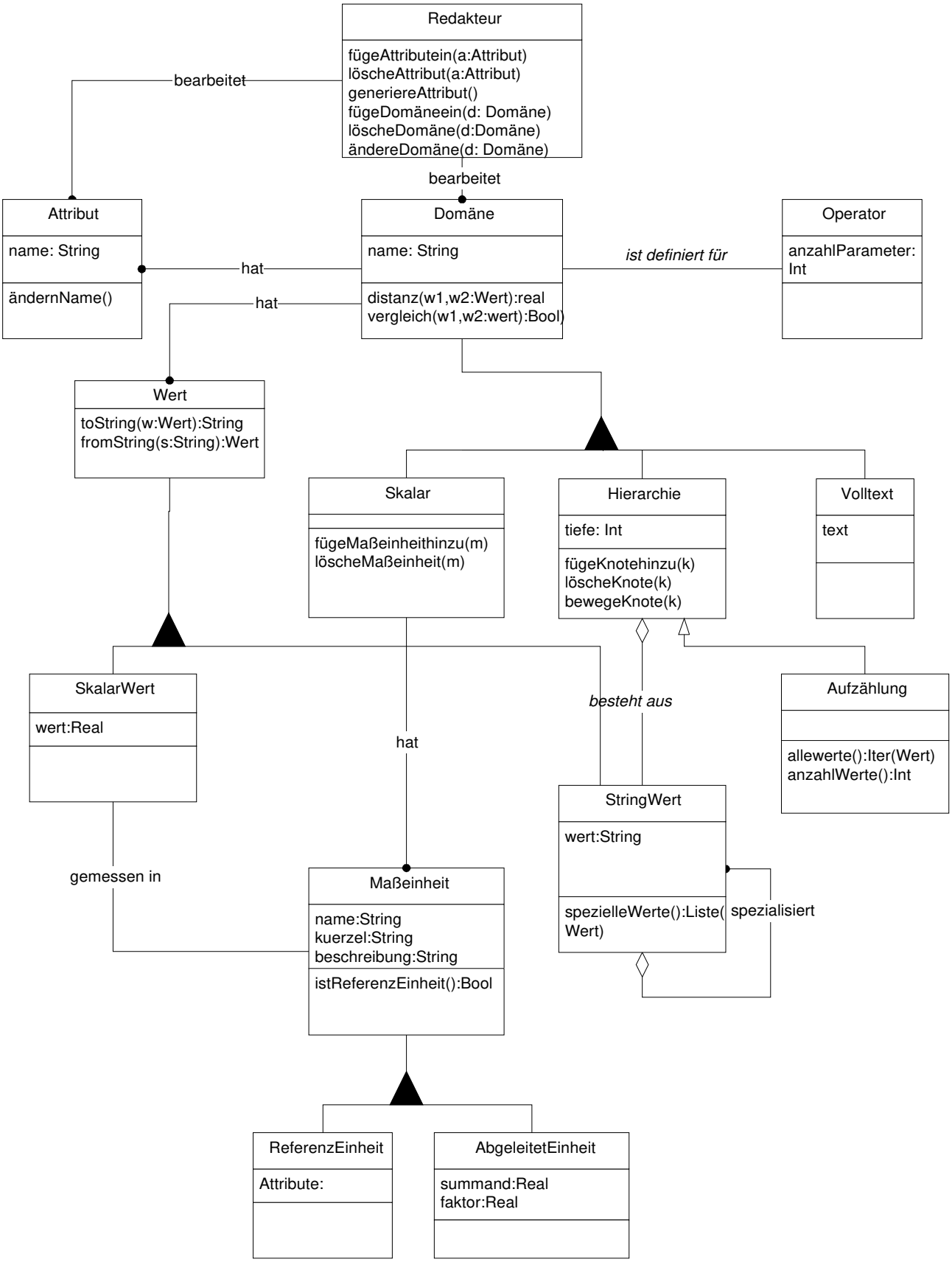

Abbildung 3.5: Die Klassen Redakteur, Domäne usw.

Die Aufgabe des Redakteurs ist die Verwaltung der Attribute und Domänen von PIA, dafür sind die folgenden Methoden definiert:

- *fügeAttributEin(a: Attribut)*
- *löscheAttribut(a: Attribut)*
- *ändereAttribut(a: Attribut)*
- *fügeDomäneEin(d:Domäne)*
- *löscheDomäne(d: Domäne)*
- *ändereDomäne(d: Domäne).*

Die Klasse Domäne hat drei Unterklassen:

- Skalar,
- Hierarchie,
- Volltext.

Im Gegensatz zu der Analysephase wird die Klasse *Aufzählung* nicht als eine der direkten Unterklassen von *Domäne* definiert, sondern als die Unterklasse der Klasse *Hierarchie*. Das ergibt sich daraus, daß sich die Objekte dieser beiden Klassen ganz ähnlich verhalten bis auf die zugehörige Benutzerschnittstelle. Das ist auch der Grund, warum die Anwendungsfälle im vorigen Kapitel im einzeln beschrieben werden. Für die Klasse *Hierarchie* ist ein Attribut namens „*tiefe*“ definiert. Durch Vererbung übernimmt die Klasse *Aufzählung* diese Eigenschaft. Ein Objekt der Klasse *Aufzählung* zeichnet sich dadurch aus, daß der Wert des Attributs „*tiefe*“ stets auf 1 gesetzt ist.

Ein Skalar besteht aus einer Menge von Skalarwerten. Jeder *Skalarwert* ist in einer bestimmten *Einheit* gemessen und die Klasse Einheit unterteilt sich wieder in *Referenzeinheit* und *AbgeleitetEinheit*. Die für die Umrechnung benötigten Daten *Faktor* und *Summand* werden als Attribute in *AbgeleitetEinheit* gespeichert.

Die Klasse Wert hat zwei Unterklassen: *SkalarWert* und *StringWert*. Ein *SkalarWert* faßt den realen Wert und die Einheit, in der der Wert gemessen ist, zusammen.

3.1.6 die Klassen Anbieter, Produkt usw.

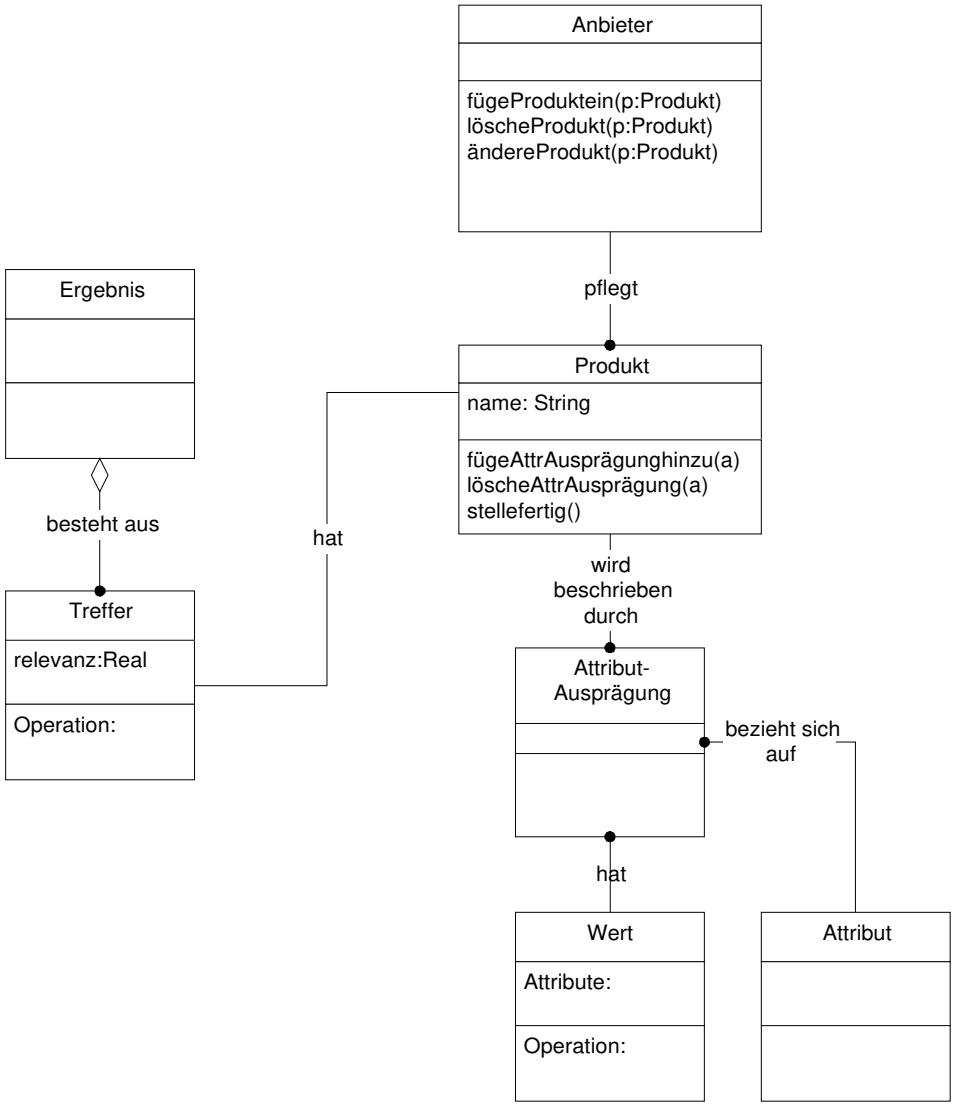

Abbildung 3.6: Anbieter, Produkt usw.

3.1.7 Die Klasse Anbieter

Ein Anbieter verwaltet seine Produktliste in PIA, dafür stehen folgenden Methoden zur Verfügung:

- *fügeProduktHinzu (p: Produkt)*
  Diese Methode nimmt ein Produkt in die Produktliste eines Anbieters auf.

- *löscheProdukt(p:Produkt)*
  Durch Aufruf dieser Methode wird das Produkt aus der Produktliste eines Anbieter gelöscht.

- *ändereProdukt(p:Produkt)*
  Diese Methode bietet dem Anbieter die Möglichkeit an, die Attribute und ihrer Ausprägungen eines von ihm eingefügten Produkts zu ändern.

3.1.8 Die Klasse Produkt und die Klasse Treffer

Das Suchergebnis in PIA besteht aus einer Menge von Treffern, wobei sich ein Produktobjekt und ein Trefferobjekt darin unterscheiden, daß letzteres mit einer reellen Zahl versehen wird, die als Relevanz bezeichnet wird. Diese Zahl wird von PIA beim Ausführen des Suchauftrages ermittelt.

Ein Produkt in PIA wird durch eine Menge von Attributausprägungen beschrieben, wobei eine Attributausprägung aus einem Attribut und einem Wert besteht.

3.2 Die Controlklasse PIA

Die Klasse PIA hat folgende Operationen:

- *meldeAktorAn(benutzername, paßwort: String): Aktor*

  Wenn ein Benutzer versucht, sich einzuloggen, stellt PIA anhand der Benutzereingabe eine Abfrage an die Datenbank und bekommt Benutzerdaten von der Datenbank zurück, indem die Methode *meldeAktoran* aufgerufen wird. Dadurch wird ein Akteursobjekt erzeugt.

- *fügeRolleHinzu(r: Rolle a: Aktor)*

  Durch eine weitere Abfrage an die Datenbank bekommt PIA eine Liste von den Rollen, die das Akteursobjekt besitzen soll und fügen die entsprechenden Rollenobjekte in das Akteursobjekt ein.

- *holeSuchaufrägen(a: Aktor): List(Suchauftrag)*

  PIA ruft dann die Methode kannSuchen() von dem Akteursobjekt, um herauszufinden, ob der Benutzer berechtigt ist, Suchaufträge zu erstellen. Bei einer positiven Antwort holt PIA die Suchaufträge des Aktors von der Datenbank mit der Methode *holeSuchaufträge*, die eine Liste von Suchaufträgen zurückliefert.

- *holeProdukte(a: Aktor) :List(Produkt)*

  Der PIA ruft die Methode *holeProdukte* auf, um die Produktliste des Aktors zu erhalten. Diese Methode kann nur aufgerufen werden, wenn der Aktor die Rolle „Anbieter“ hat.

- *führeSuchauftragaus(s:Suchauftrag): List(Produkt)*

  PIA führt einen Suchauftrag aus, indem er eine Suchanfrage aus dem Suchauftragobjekt generiert und sie an die Datenbank stellt. Als Rückgabe bekommt er eine Liste von Produkten zurück.

- *berechneRelevanz (p:Produkt s:Suchauftrag) : Real*

Für jedes Produkt, das PIA von der Datenbank zurückbekommt, rechnet er eine reelle Zahl aus. Diese reelle Zahl sagt aus, inwiefern das Produkt diesem Suchauftrag entspricht.

- *berechneAttrRelevanz (a: Attribut p:List(Produkte)): Real*

  Für jedes Attribut rechnet PIA die Häufigkeit aus, mit der ein Attribut in einer Liste von Produkten auftaucht.

- *zähleTreffer(s: Suchauftrag r: Reel) : Int*

  Diese Methode liefert die Anzahl der Treffer, deren Relevanz größer als w sind.

- *holeGemeisamenAttribute(e:Ergbenis): List(Attribut)*

  Diese Methode liefert aus dem Suchergebnis – einer Liste von Produkten – die gemeinsamen Attribute zurück.

- *speichereSuchauftrag(s:Suchauftrag,a:Aktor)*

  Durch Aufruf dieser Methode wird dieser Suchauftrag von dem gegebenen Akteursobjekt in der Datenbank gespeichert.

- *löscheSuchauftrag(s:Suchauftrag,a:Aktor)*

  PIA löscht diesen Suchauftrag des gegebenen Akteursobjekts aus der Datenbank.

- *sortiereAttribute (a: List(Attribut) s:Suchauftrag): List(Attribut)*

  Jedesmal wenn eine Attributanfrage erzeugt wird, wird sie zu dem Suchauftragobjekt hinzugefügt und dadurch wird das Suchauftragobjekt aktualisiert. PIA ermittelt das Ergebnis von diesem „Zwischensuchauftrag“ und liefert eine Liste von Produkten zurück. Daraufhin wird die Methode *berechneAttrRelevanz(a:Attribut l: Liste(Produkt)):Real* für jedes Attribut des Systems einmal aufgerufen und die Attributliste absteigend nach dieser Relevanz neu sortiert.

3.3 Java Interface Klassen

Die graphische Benutzeroberfläche stellt eine der wichtigsten Komponente eines Informationssystems dar. Sie ist die Schnittstelle zum menschlichen Benutzer und sollte die Kommunikation so einfach wie möglich gestalten. Für das Frontend des PIA wird Java als Implementierungssprache ausgewählt.

3.3.1 Warum Java

Was in der Evolution des WWW bis vor kurzem gleich geblieben war, ist der Ansatz, daß der Web-Browser nur ein Frontend ist und alle Programmlogik, Verarbeitung oder auch nur Eingabeprüfung auf der Seite des Servers stattfindet. Die Idee, anstelle von HTML-Dateien

auch Programmcode zum Browser zu übertragen, der dort ausgeführt wird, stellt folgende Anforderungen an den Programmcode sowie die Umgebung, in der dieser Code ablaufen soll:

- Der Programmcode muß plattformunabhängig sein, weil es nicht vorausgesagt werden kann, auf welcher Plattform der anfragende Browser läuft. Das heißt, der Programmcode soll nicht irgendeine spezielle Maschinencode sein soll eine neutrales Format besitzen.
- Die Ausführung des Programmcodes soll effizient sein. Da es schon einige Zeit kostet, den Programmcode über das Netzwerk zu verschicken, soll zumindest die Ausführung performant sein.
- Damit das Frontend in einer verteilten Umgebung wie dem Internet sinnvoll funktioniert, muß besonders viel Wert auf die Berücksichtigung von sicherheitsrelevanten Features gelegt werden.
- Der Programmcode muß einem aktuellen Paradigma folgen, um sich zu verbreiten und durchzusetzen.

3.3.2 Was ist Java

In „The Java Language: A White Paper“ beschreibt Java seiner Erfinder Sun als:

> A simple, object-oriented, distributed, interpreted, robust, secure, architecture neutral, portable, high-performance, multithreaded, and dynamic language.

Java ist eine einfache Sprache, die man sehr schnell lernen kann. Ihre Entwicklung begann 1990 bei der Firma Sun in einem Team. Durch ihre Portabilität, die eine Kernforderung bei der Konzeption der Sprache war, ist Java ideal für die verteilte Anwendung in Internet. Außer der Portabilität hat Java noch folgende wichtigen Eigenschaften, die Java auszeichnet:

- 100% objektorientiert,
- multithreaded,
- verteilt,
- robust und
- sicher.

3.3.3 Interface Klassen

Interfaceklasssen sind diejenigen Klassen, die für die Darstellung der Informationen und die Interaktionen mit dem Anwender bestimmt sind. Beim Zusammenstellen eines neuen Suchauftrages werden dem Benutzer die sämtlichen Attribute der Datenbank aufgelistet. Der Benutzer darf ein beliebiges Attribut auswählen und dessen Ausprägung mit Hilfe eines AA-Editors (vgl. Abb. 3.6) festlegen. Jedes ausgewählte Attribut wird zusammen mit seiner vom Benutzer festgelegten Ausprägung in ein zweites Dialogfenster eingetragen.

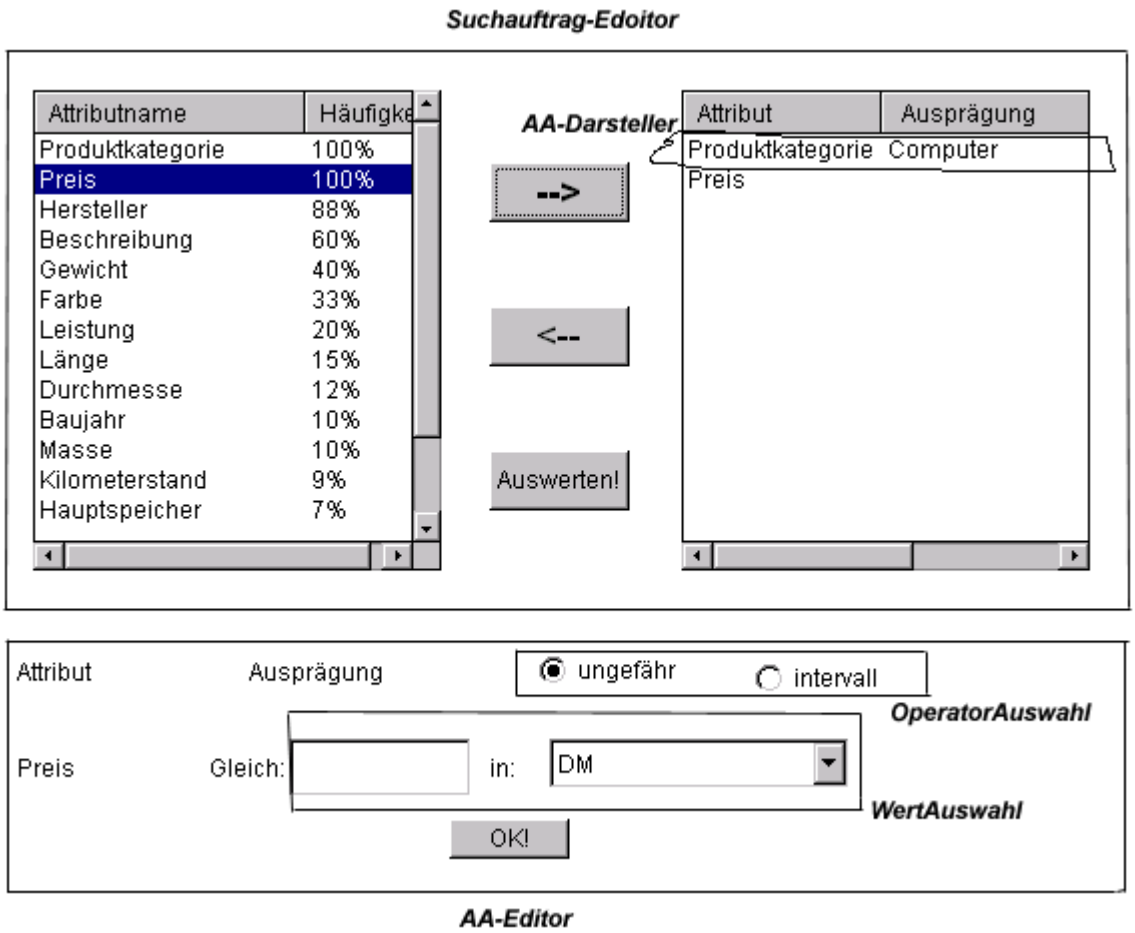

Abbildung 3.7: Interfaceklassen für Erstellung eines neuen Suchauftrages

3.3.3.1 Interfaceklassen in Überblick

Im folgenden Diagramm werden die Interfaceklassen, die für den Anwendungsfall „Produkt suchen“ benutzt werden, mit den Objektklassen, die sie darstellen, gegenübergestellt.

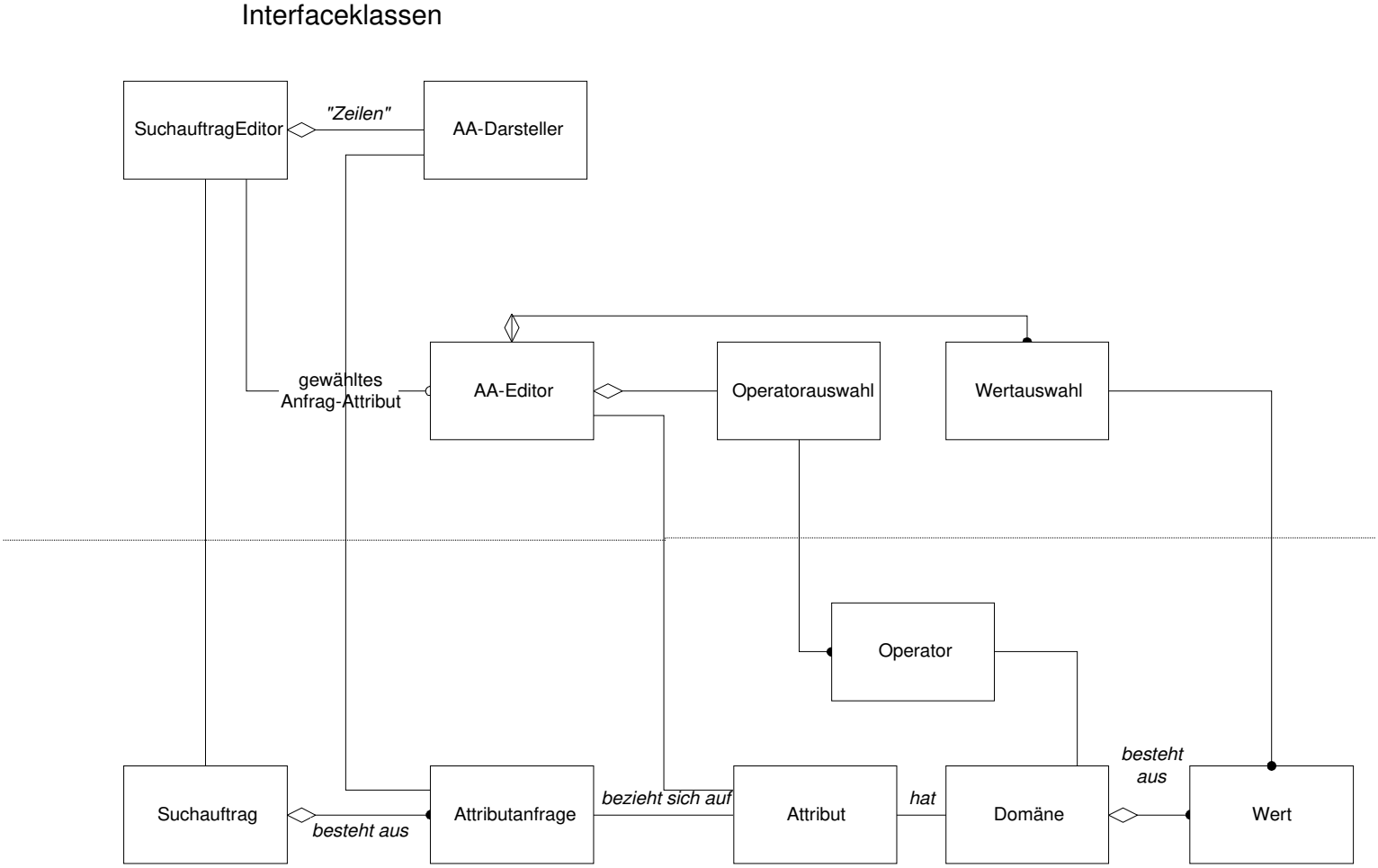

Abbildung 3.8: Interfaceklassen in Überblick

3.3.3.1.1 AA[6]-Darsteller:

Ein AA-Darsteller stellt ein Objekt der Klasse Attributanfrage auf der Benutzerschnittstelle dar.

Die Methoden:

toString(a: Attributanfrage) :String

Diese Methode wandelt ein Attributanfrageobjekt in Text um, z.B. „Preis ist zwischen 1000,00 und 2000,00 DM“.

3.3.3.3.2 SuchauftragDarsteller

[6] AA: Attributanfrage

Ein SuchauftragDarsteller stellt den Zustand eines Suchauftrags dar. Ähnlich wie die Beziehung zwischen der Klasse Suchauftrag und der Klasse Attributanfrage, besteht ein SuchauftragDarsteller aus mehreren AA-Darstellern.

Die Methoden:

- *fügeAAhinzu(a: AA-Darsteller): SuchauftragDarsteller*
- *löscheAA(a: AA-Darsteller): SuchauftragDarsteller*

3.3.3.3.3 SuchaufträgListenDarsteller

Diese Klasse stellt die eine Liste von Suchaufträgen eines Kundenobjektes dar.

3.3.3.3.4 AA-Editor:

Je nach dem, welches Attribut der Benutzer ausgewählt hat, erzeugt ein AA-Editor eine domänenspezifische Benutzerschnittstelle, womit der Benutzer den Operator und Attributausprägungen für dieses Attribut festlegen kann. Ein AA-Editorobjekt enthält je eine Referenz auf ein Operatorauswahlobjekt und ein Wertauswahlobjekt.

Ein Operatorauswahlobjekt enthält mehrere Operatoren und stellt sie als ein Gruppe von Radiobuttons dar. Jeder Radiobutton ist mit einem Operator, der für die Domäne definiert ist, beschriftet.

Je nach dem, welchen Operator ausgewählt wird, zeigt der AA-Editor die entsprechende Eingabeschnittstelle an, wo der Benutzer Werte angeben kann, mit denen er dieses Attribut belegen möchte.

3.3.3.3.5 TrefferDarsteller

Die Treffer einer Suchanfrage werden in tabellarischer Form dargestellt. Ein Trefferdarsteller stellt einen Treffer in zwei Tabellenzeilen dar. In der ersten Zeile stehen die Attributausprägungen der gemeinsamen Attribute aller Treffer einschließlich der Wahrscheinlichkeitszahl, und zwar jede Ausprägung in einer Tabellenspalte. In der zweiten Tabellenzeile werden die weiteren Angaben zu dem Produkt als Text dargestellt.

3.3.3.3.6 ErgebnisDarsteller

Ein Ergebnisdarsteller besteht aus einer Menge von Trefferdarstellern und dem Tabellenkopf, in dem die gemeinsamen Attribute repräsentiert werden.

3.3.3.3.7 AttributListeDarsteller

Ein AttributListeDarsteller stellt eine Liste von Attributen dar, die ihm übergeben wird.

3.4 Interaktionen zwischen den Objektklassen und Interfaceklassen

3.4.1 Das Muster Model-View-Controller (MVC)

Die Grundidee des MVC besteht in der Trennung der zum Problembereich gehörenden Objektklassen von den Interfaceklassen.

Eine Zustandsänderung in einem Entity-Objekt resultiert aus dem Empfang einer Nachricht von demselben oder einem anderen Objekt (z.B. von einem gegenwärtig aktiven Interface-Objekt, einem anderen Entity-Objekt, einem Controlobjekt usw.) Um diese Änderungen in allen verwandten Interface-Objekten widerzuspiegeln, verwendet das MVS einen speziellen Verarbeitungsmechanismus.

Dieser beruht darauf, daß jedes Entity-Objekt eine Liste abhängiger Objekte besitzt, in der alle verwandten Interface-Objekte registriert sind. Wenn sich der Zustand eines Entity-Objekts ändert, wird eine Nachricht an alle abhängigen Objekte gesendet, um diese von der Änderung in Kenntnis zu setzen. Die verwandten Interfaceobjekte fragen den Zustand des Entity-Objekts ab und aktualisieren ihre Anzeige entsprechend.

Das folgende Objektinteraktionsdiagramm illustriert diesen Verarbeitungsmechanismus.

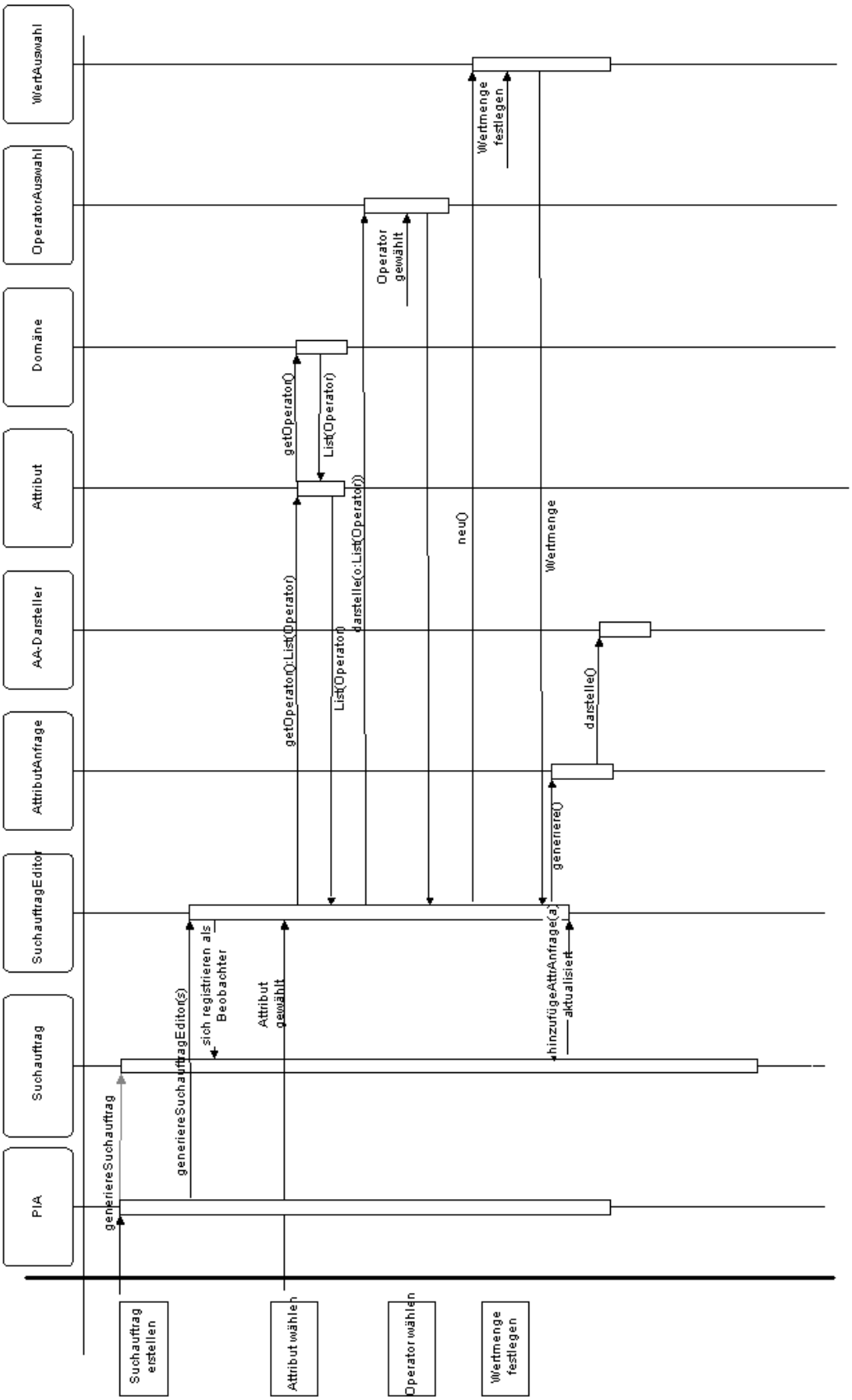

Abbildung 3.9: Interaktionsdiagramm zwischen den Objekten

Bei der Erstellung eines neues Suchauftrags erzeugt PIA zuerst ein neues Suchauftragobjekt, das zunächst keine Attributanfrage enthält. Dann wird ein SuchauftragEditorobjekt erzeugt mit dem zuvor erzeugten Suchauftragobjekt als Parameter, so daß das Editorobjekt den Suchauftrag richtig darstellen kann. Damit der Suchauftrageditor von den Zustandsänderungen des Suchauftrags informiert wird, registriert sich der Suchauftrageditor bei dem Suchauftrag als ein Beobachter.

Wenn ein Benutzer ein Attribut ausgewählt hat, fragt das SuchauftragEditorobjekt das Attributobjekt, welche Operatoren definiert sind. Das Attribut-Objekt leitet diese Anfrage an das zugehörige Domänenobjekt weiter. Es erhält eine Liste von passenden Operatoren. Mit dieser Operatorliste wird dann ein OperatorAuswahlobjekt erzeugt und angezeigt.

Nachdem der Benutzer einen Operator auswählt hat, wird ein WertAuswahlobjekt erzeugt, das entweder Eingabefelder oder Pull-Down-Menü enthält. Dort kann der Benutzer die Werte festlegen. Nachdem das Attribut, der Operator und die Werte festgelegt sind, kann eine Attributanfrage erzeugt werden, die dann direkt zu dem Suchauftrag hinzugefügt wird. Wenn das Suchauftragobjekt die neue Attributanfrage in seine Liste aufgenommen hat, muß er seine Beobachter über den Zustand informieren, damit die Beobachter immer den aktuellen Zustand darstellen. Nach Erhalten dieser Nachricht fragt der Beobachter (in diesem Fall der SuchauftragEditor) den Zustand dieses Suchauftrags ab und aktualisiert entsprechend die Darstellung.

Kapitel 4

Zusammenfassung und Ausblick

Die vorliegende Arbeit beschreibt die Analyse und den Entwurf eines PIA Systems. Dabei wird ein objektorientierte Ansatz der Softwareentwicklung verfolgt, der aus den drei Phasen Analyse, Entwurf und Implementierung besteht. In dieser Arbeit wird ein Analysemodell entwickelt, in dem zuerst die Anwendungsfälle des Systems beschrieben werden. Auf der Basis dieses Analysemodells wird ein Objektmodell aufgebaut, das die gesamte Struktur des Systems darstellt. Daraufhin wurde beispielhaft ein Java-Frontend implementiert, das auf der Messe NORTEC vorgeführt wurde.

Da in dieser Arbeit in erster Linie auf die Analyse und Entwurf Wert gelegt wird, bleiben noch viele technischen Details und ein großer Teil der Implementierung offen. Dazu gehören:

- Auswahl von Datenbanksystemen und konkretes Datenmodell,

  In Frage kommen ein RDDBMS oder OODDBMS, die jeweils ihre Vor- und Nachteile haben. Auf die Beschreibung eines konkreten Datenmodells wird hier verzichtet.

- Anbindung der Datenbank,

  Je nachdem, welches Datenbanksystem gewählt wird, wird entsprechend die Datenbankanbindungsmöglichkeit gewählt, sei es über eine herstellerspezifische API oder eine standardisierte Schnittstelle wie ODBC, JDBC oder DBI usw.

- Aufbau 3-Tier-Architektur,

  Es spricht vieles dafür, PIA in einer 3-Tier-Architektur aufzubauen. Der Benutzer interagiert mit dem Java-Fontend, während Suchalgorithmus auf dem Applikationsserver durch Zugriff auf die Datenbank den Suchauftrag erledigt.

- Kommunikation zwischen Client und Applikationsserver.

  In einer verteilten Anwendung wie PIA ist die Kommunikation zwischen Client und Server sehr wichtig. Um sowohl ein herstellerspezifisches und daher nicht offenes Protokoll als auch eine nur auf syntaktischer Ebene aufbauende Socketverbindung zu vermeiden, sollte eine moderne Technik wie Corba eingesetzt werden, die eine Reihe von Vorteilen mit sich bringt: Interface-Definition, verschiedene Sprache-Mapping, Standard-Service usw.

- Implememententierung eines Suchalgorithmus.

  Der schwierigste Teil des PIA dürfte die Implementierung des generischen, universellen, auf semantischer Ebene basierenden Suchalgorithmus sein.

- Fertigstellung des kompletten Java-Frontendes.

Die Implementation, die im Rahmen dieser Arbeit implementiert wurde, zeigt nur die Möglichkeit der Programmiersprache Java und gibt auch einen Vorgeschmack auf das gesamte System, die noch nicht alle im System vorhandenen Anwendungsfälle abdeckt.

Literaturverzeichnis

Rumb 93 J. Rumbaugh, M. Blaha, W. Premerlani, F. Eddy, W. Lorensen. *Objektorientiertes Modellieren und Entwerfen.* Hanser Verlag, 1993.

Gamma 96 E. Gamma, R. Helm, R. Johnson, J. Vlissides. *Entwurfsmuster*. Addison-Wesley, 1996.

Midd 96 S. Middendorf, R. Singer, S. Strobel. *JAVA Programmier Handbuch.* Dpunkt, 1996.

Yourdon 96 E. Yourdon. *Mainatream Objects*. Prentice Hall Verlag, 1996.

Biskup 95 J. Biskup. *Grundlagen von Informationssystemen*. Vieweg, 1995.

Meyer 90 B. Meyer. *Objektorientierte Softwareentwicklung.* Hanser, 1990.

Fuhr96 N. Fuhr. *Models for integrated Information Retrieval and Database.* 1996.

Flag 96 D. Flanagan. *JAVA in a Nutshell.* 1996, O‘ Reilly & Associates.

Iiva 95 J. Iivari, K. Lyytinen, M. Rossi. *Advanced Information Systems Engineering.* Spinger Verlag, 1995.

Patyra 96 M. J. Patyra, D. M. Mlynek. *Fuzzy-Logic Implementation and Applications.* Wiley Teubner, 1996.

Amin 94 F. Aminzadeh, M. Jamshidi. *Soft Computing-Fuzzy Logic, Neural Networks, and Distributed Artificial Intelligence.* Prentice Hall, 1994.