

# **CORBA-basiertes Workflow-Management zur Bearbeitung von Kundenanfragen aus dem Internet**

Felix Meyer<sup>1</sup>, Matthias Reisecker<sup>2</sup>, Falk Krebs<sup>2</sup>

<sup>1</sup> Daimler-Benz Forschung und Technologie, Software-Technologie – Prozesse und Qualität  
felix.meyer@dbag.ulm.DaimlerBenz.com

<sup>2</sup> Fachhochschule Ulm - Fachbereich Technische Informatik

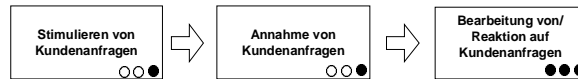
**Abstract.** Electronic Commerce verändert die Kommunikation zwischen Unternehmen und Kunden. Der Web Server eines Unternehmens löst den klassischen Ladentisch ab. Bei Kundenanfragen, die über "Customer Self Service" hinausgehen, ist eine Einbindung des Kunden in interne Arbeitsabläufe des Unternehmens erforderlich [Bishop et al. 97]. Dieser Beitrag zeigt mit dem "Customer Interaction Management System" einen Ansatz zur strukturierten Bearbeitung von Kundenanfragen mit dem Ziel einer schnellen und kompetenten Beantwortung. Erreicht wird dies durch eine leistungsfähige Unterstützung der kooperativen Arbeitsprozesse, die mit der Findung der Experten und der Beantwortung der Anfragen verbunden sind. Das vorgestellte System basiert dabei auf einer CORBA Architektur, der sich sowohl der externe Kunde als auch die internen Experten bei der Beantwortung der Anfrage bedienen. Die offene Gestaltung des Systems erlaubt zum einen eine durchgängige Benutzeroberfläche für Kunden und Experten und zum anderen eine unkomplizierte Erweiterung des Systems und Nutzung der gewonnenen Daten durch dritte Systeme.

## **1 Einleitung**

Zunächst soll die Thematik „Customer Interaction Management“ im Zusammenhang betrachtet werden, um die sich daraus ergebende Funktionalität für das realisierte "Customer Interaction Management System" zu begründen.

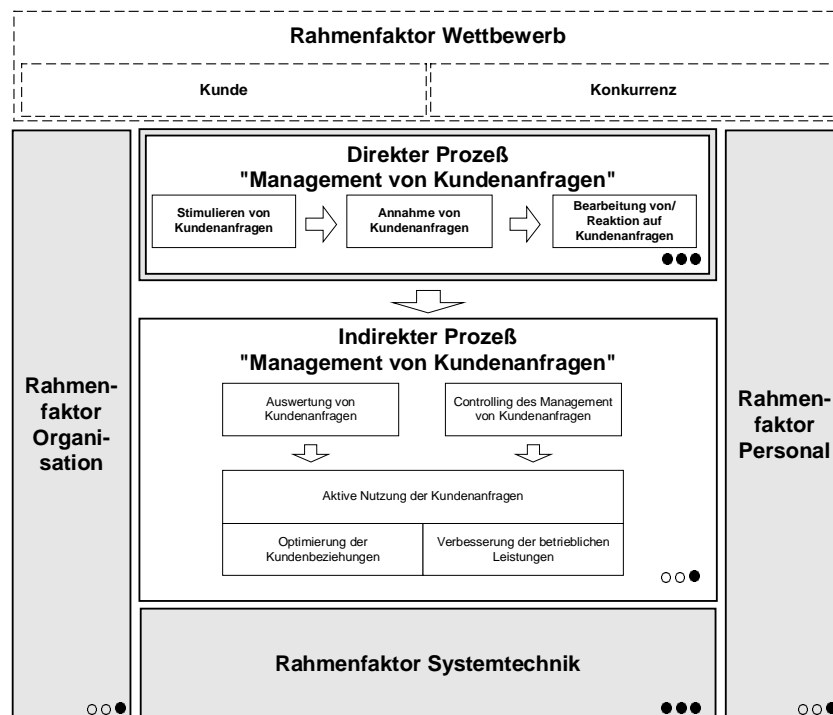
### **1.1 Management von Kundenanfragen**

Geht man von einem systematischen Management von Kundenanfragen aus, so läßt sich in Anlehnung an die Systematik des „Aktiven Beschwerdemanagements“ [Stauss & Seidel 95] der eigentliche Prozeß „Management von Kundenanfragen“ in drei Schritte unterteilen (Abb. 1). Die Anzahl der schwarzen Kreise (●) verdeutlicht, wo die Schwerpunkte dieses Beitrags liegen.



**Abb. 1.** Prozeß „Management von Kundenanfragen“

Dieser Prozeß „Management von Kundenanfragen“ umfaßt nicht die weitere Beurteilung und Auswertung der Anfragen für das eigene Unternehmen. Man unterscheidet daher den *direkten* und den *indirekten* Prozeß des Managements von Kundenanfragen (Abb. 2).



**Abb. 2.** Aufgabenspektrum des Managements von Kundenanfragen

Der direkte Prozeß beinhaltet immer die Interaktion mit dem Kunden, während der Kunde an dem indirekten Prozeß nicht mehr beteiligt ist. Der indirekte Prozeß birgt großes Potential für das Unternehmen, da er die Chance zur Verbesserung der Kundenbeziehungen und der eigenen betrieblichen Leistungen bietet.

Beide Prozeßtypen werden stark von Rahmenbedingungen des Unternehmens beeinflußt. Daher sind bei einer Behandlung der Prozesse diese Rahmenfaktoren ebenfalls zu berücksichtigen. Ein wesentlicher Rahmenfaktor ist auch der Wettbewerb. Allerdings ist er als starr und vom Unternehmen nicht zu beeinflussen anzusehen.

## 2 Konzept des Customer Interaction Management Systems

Das Konzept des Customer Interaction Management Systems unterstützt jeden Prozessschritt des direkten Managements von Kundenanfragen (Abb. 3). Ein zentraler Punkt ist dabei die Expertenzuordnung, die statisch oder dynamisch geschehen kann.

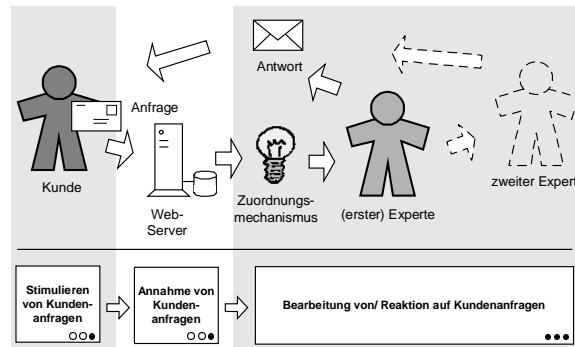
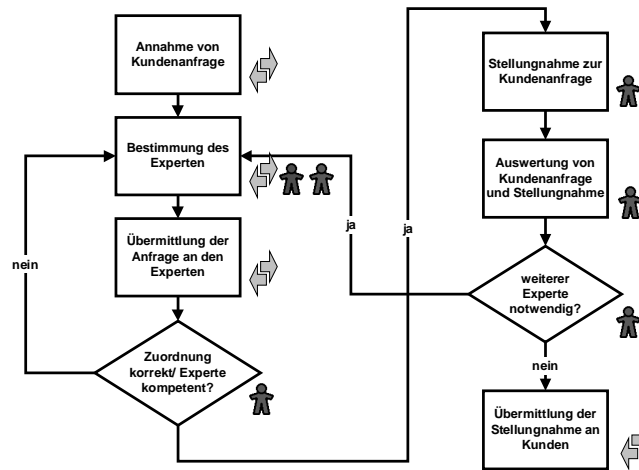


Abb. 3. Direkter Prozeß

Grundlage für eine statische Zuordnung können Inhalte der Web-Seite oder Angaben über den Kunden sein. Dynamische Zuordnung kann automatisch - z.B. durch Schlüsselwortsuche - geschehen oder per Hand vorgenommen werden. Obwohl dynamisch-automatische Zuordnungsmechanismen bei der Expertenfindung eine wichtige Rolle spielen können, sind diese nicht Teil des CIMS Konzeptes. Vielmehr können sie als externe Implementierungen in das CIMS eingebunden werden. Prinzipiell ist das CIMS so flexibel ausgelegt, daß es alle Methoden der Expertenfindung unterstützen kann. Der Zuordnungsmechanismus kann beliebig und auch in Kombination mit einem anderen gewählt werden. Dies ist ein wesentlicher Vorteil gegenüber anderen Systemen. Hauptaufgabe des Customer Interaction Management Systems ist allerdings nicht die Expertenfindung sondern die optimale Unterstützung der kooperativen Bearbeitung von Anfragen im Sinne eines einfachen Workflow Management Tools. Da der Bearbeitungsprozeß nicht von vornherein klar bestimmt ist, soll es den Bearbeitungsprozeß flexibel unterstützen. Dabei basiert das CIMS von der Erfassung der Anfrage bis zur Antwort durch den internen Experten komplett auf Internet/ WWW-Technologie. Alle Eingaben und Aktionen erfolgen über einen Web-Browser.

### 2.1 Prozeßmodell

Die Bearbeitung von Kundenanfragen folgt unterstützt durch das CIMS dem in Abb. 4 dargestellten Prozeßmodell. Der Prozessschritt „Bestimmung des Experten“ kann dabei automatisch oder manuell geschehen. Bei der manuellen Expertenfindung bietet das CIMS Möglichkeiten zur Weiterleitung der Anfragen an Experten.



**Abb. 4.** Prozeßmodell des Customer Interaction Management Systems

Abb. 4 zeigt, daß außer der manuellen Expertenfindung weitere dynamische Anteile des Arbeitsablaufes den Workflow flexibel machen. Dies sind die beiden Entscheidungen „Zuordnung korrekt/ Experte kompetent?“ und „weiterer Experte notwendig?“. Hier liegt es am jeweiligen Experten, wie die weitere Bearbeitung der Anfrage geschieht.

## 2.2 Expertenstruktur

Teil des CIMS ist die Expertenstruktur, die es überhaupt erst möglich macht, Arbeitsabläufe flexibel zu gestalten. Die Expertenstruktur wächst mit der Dauer und Anzahl der bearbeiteten Anfragen. Sie spiegelt das Wissen des Unternehmens abgebildet auf Personen bzw. Rollen wider. Während der Anwendung des CIMS protokolliert das System die verschiedenen Stationen des Workflows und speichert sie in einer Datenbank. So läßt sich später gut nachvollziehen, welcher Experte zu welchem Themengebiet bereits konsultiert wurde und für eine erneute Bearbeitung einer Anfrage zum gleichen Themengebiet in Frage kommt. Umgekehrt ist genauso der Ausschluß bestimmter Personen möglich, die Anfragen zu dem Themengebiet bereits zuvor zurückgewiesen haben. Die Assoziation zwischen Experte und Themengebiet ist ebenso zwischen Experte und Kunde oder Experte und statischer Web-Seite möglich.

## 2.3 Alleinstellungsmerkmale

Das Konzept des Customer Interaction Management Systems greift die gegenwärtige Problematik vieler kommerzieller Web-Sites auf, die Kundenanfragen nicht die Bedeutung schenken, die sie benötigen [Meyer 97]. Kundenanfragen dürfen nicht nur als Pflicht für ein Unternehmen gesehen werden, sondern stellen eine erhebliche

Chance dar, über Rückmeldungen der Kunden die eigenen Produkte und Prozesse zu verbessern [Gaitanides et al. 94].

Gegenüber anderen Ansätzen weist das Konzept des CIMS einige wesentliche Alleinstellungsmerkmale auf:

1. Die strikte **Verwendung von Standards** ermöglicht den Einsatz des CIMS als Erweiterung bestehender Web-Systeme auf quasi allen Plattformen. Bei späterer technologischer Erweiterung des Systems ist der Aufwand geringer als bei Verwendung proprietärer Software.
2. Die **Erfassung und spätere Verfügbarkeit der Web-Seite**, von der die Kundenanfrage ausging, ist als Hilfestellung für die kompetente Beantwortung der Anfrage durch den Experten eine wesentliche Funktionserweiterung. Sie beugt vielen einfachen Mißverständnissen vor und erlaubt zudem eine Klassifizierung der Experten.
3. Die **Strukturierung von Experten bzw. deren Wissen** ist zum einen wichtig für die Verteilung der Last bei einer großen Anzahl von Anfragen und zum anderen für die schnelle Verfügbarkeit bei seltenen und ausgefallenen Anfragen. Sie trägt damit sowohl zur Expertise der Antworten als auch zur Verringerung der Bearbeitungszeit einer Kundenanfrage bei.
4. Viele Feedbacksysteme verwenden starre Zuordnungsmechanismen, die bei Schwankungen der Anfragelast oder ineffizienten Zuordnungsergebnissen nur schwer geändert werden können. Durch die Möglichkeit der **dynamischen Veränderung des Zuordnungsmechanismus** von Anfrage zu Experten ist die Optimierung des Gesamtsystems zu jeder Zeit möglich.

Insgesamt kann das Customer Interaction Management System einen wesentlichen Beitrag zum individuellen Service, den das Unternehmen seinen Kunden bietet, liefern. Gerade vor dem Hintergrund der internationalen Verfügbarkeit des Internets und der damit zwangsläufig auftretenden Konkurrenz ist individueller Service ein wichtiger Faktor im Bemühen um Kunden im Electronic Commerce [Frese & Noetel 92].

### **3 Werkzeuge und Techniken zur Implementierung des Prototypen**

#### **3.1 Programmierungsumgebung**

Das CIMS stellt einige Forderungen an die zu verwendende Programmiersprache. Netzwerke sollen auf hoher Ebene unterstützt werden und der erzeugte Code soll möglichst portabel sein. Weitere wichtige Aspekte sind Sicherheitslösungen, da sich das CIMS im Umfeld Internet/ WWW bewegt und die Unterstützung von Threads. All diese Kriterien werden von Java erfüllt [Flangan 98]. Im CIMS wurde das Java JDK 1.1.3 zur Entwicklung eingesetzt.

### 3.2 Servlets

Ein wichtiger Punkt für den universellen Einsatz des Systems ist die Notwendigkeit, daß die Clients (Kunden- und Expertensysteme) im Zugriff auf das System von lokalen Softwareinstallationen befreit bleiben. Erste Überlegungen beschäftigten sich mit der Verwendung von Applets. Jedoch wurde der Einsatz von Applets im Hinblick auf Downloadzeiten, Unterschiede in den Implementierungen der JVM von Browsern und dem Sandbox-Prinzip (für DB-Anwendungen nicht sonderlich geeignet) verworfen.

Als interessante Alternative zu Applets erwies sich das Gegenstück zu den clientseitigen Bytecode-Objekten: Servlets. Servlets besitzen die gleiche Funktionalität wie eine Java-Applikation (mit Ausnahme des graphischen Interface) und vereinen zusätzlich die Konzepte von Applets und CGI. Der wesentliche Vorteil gegenüber konventionellem CGI besteht darin, daß Servlets dynamisch vom Server geladen werden können und über die Dauer einer einzelnen Abfrage hinaus auf dem Server bestehen können. Dadurch ergibt sich gegenüber einer CGI-Lösung ein respektable Geschwindigkeitsvorteil [Orfali & Harkey 98]. Weiterhin ist es bei Servlets nicht notwendig, die HTTP-Methode (Get, Post) zu identifizieren, um die übergebenen Parameter zu verarbeiten. Mit diesen Einzelheiten beschäftigt sich der Web-Server. Im Falle von CGI muß dem Programmierer die Übertragungsmethode bekannt sein oder er muß die Methode per Programmierung entschlüsseln und danach die entsprechende Verarbeitung durchführen.

Mit dem Vorteil der serverseitigen Datenverarbeitung minimiert sich der clientseitige Softwareaufwand auf die Benutzung von JavaScript (Validierung der Eingaben des Kunden, Zusammenfassen von Eingaben) und HTML.

### 3.3 JDBC

Im CIMS besteht die Notwendigkeit die verarbeiteten Daten (Kundenanfrage, Expertise, Expertenstruktur, Sekundärinformation, Verwaltungsdaten) strukturiert abzuspeichern. Hierfür wurde eine Microsoft SQL Datenbank installiert. Im Zuge der Plattformunabhängigkeit, die das CIMS kennzeichnet, und hinsichtlich Java bietet sich die Verwendung von JDBC für den Zugriff auf die Daten an [Reese 97]. Die Plattformunabhängigkeit kann nur die Verwendung eines JDBC-Treiber des Typ 4 [Sun] garantieren, der in reinem Java-Code implementiert ist. Das JDBC-Paket setzt auf den Java-Basisklassen auf und stellt relationale Datenbankobjekte und die entsprechenden Methoden für den Zugriff aus Java-Applikationen auf beliebige Datenbanken bereit.

Der oben erwähnte JDBC-Treiber wird beim JDBC-Treiber-Manager angemeldet. Beim JDBC-Treiber-Manager können mehrere JDBC-Treiber parallel angemeldet werden und somit der Zugriff auf unterschiedliche Datenbanken realisiert werden. Diese Datenbanken können sich auf beliebigen Rechner befinden, da beim Erzeugen einer Verbindung eine URL angegeben wird. Somit ist eine Verteilung der Rechnerlast auf einfache Weise möglich.

### 3.4 CORBA

Das System kann mit den oben geschilderten Mitteln wichtige Aspekte im Bezug auf die Plattformunabhängigkeit und die Verteilung von Rechnerauslastung (Datenbanken) erfüllen. Das Kernsystem zur Verwaltung der Anfragen ist jedoch auf einen Rechner begrenzt. Dieses Kernsystem soll im Hinblick auf zukünftige Organisationsstrukturen als Client/ Server-System implementiert werden. Zur Realisierung dieser Forderung würden sich RMI oder CORBA [Vogel & Duddy 97] anbieten. Einerseits ist RMI besser in die Sprache Java integriert, was sich vor allem in der einfacheren Parameterübergabe und in der Schnittstellendefinition als Vorteil herausstellt. Andererseits verhindert diese Integration die Kommunikation zwischen Objekten verschiedener Programmiersprachen (Homogenität), was zusätzlich durch das RMI-spezifische RMI Wire Protocol (Bestrebungen RMI über IIOP zu betreiben werden von Sun angestrebt) verstärkt wird. Im Gegensatz zu Corba fehlen bei RMI jegliche Ansätze zur dynamischen Objektfindung und Vergleichbares zu den Corba Services. Daraus resultierte die Entscheidung Corba im CIMS einzusetzen.

Zentraler Bestandteil der Corba-Architektur ist der ORB, der als Nachrichtenvermittler zwischen Client und Server dient. Durch die Verteilung der Anwendungs-Objekte auf andere Rechner ergibt sich eine weitere Möglichkeit der Skalierbarkeit, was im besonderen durch den Naming Service unterstützt wird. Desweiteren eröffnet sich anderen Abteilungen die Möglichkeit, Zugang zu den Corba-Objekten zu erlangen und deren Dienste für sich nutzbar zu machen. Für diese Nutzer besteht darüber hinaus nicht einmal die Notwendigkeit, den gleichen ORB, die gleiche Programmiersprache oder die gleiche Entwicklungsumgebung einzusetzen.

Das CIMS verwendet den Corba-Bus JavaIDL der Firma JavaSoft. Im JDK 1.2 ist dieser fester Bestandteil. JavaIDL entspricht dem CORBA/ IIOP-Standard und beinhaltet volles IDL-to-Java language mapping. IIOP (beruht auf TCP/ IP) ist das von JavaIDL genutzte Übertragungsprotokoll. JavaIDL verwendet IIOP sowohl in der Interoperabilität mit ORBs anderer Hersteller als auch zur Kommunikation zwischen den erstellten Corba-Clients und Corba-Servern.

In der CIMS-Implementierung wird der Nameservice von JavaSoft benutzt. Bei diesem Service melden sich die Server-Objekte an. Die Clients erhalten über diesen die Objekt-Referenzen, mit denen sie völlig transparent Methoden des Servers nutzen. Die Aufrufe der Clients geschehen über die vom IDL-to-Java-Compiler bereitgestellten Proxy-Objekte (Stubs). Auf der Serverseite übernehmen ebenfalls Proxy-Objekte (Skeletons) eine ähnliche Funktion. Die Proxy-Objekte des jeweiligen Objekte haben dieselbe Schnittstelle implementiert wie das zugehörige entfernte Objekt. Die Proxy-Objekte konvertieren – unsichtbar für das entsprechende Objekt – die Aufrufparameter in ein plattformunabhängiges Format. Die Möglichkeiten der dynamischen Objektfindung über das DII (Dynamic Invocation Interface) ist im CIMS nicht implementiert.

## 4 Customer Interaction Management System

Der Prototyp des Customer Interaction Management Systems wurde von der Konzeption bis zur Realisierung von 3 Personen über 9 Monate entwickelt. Die Arbeiten wurden nach konzeptioneller Vorarbeit innerhalb von 2 Praxissemestern und 2 Diplomarbeiten zuzüglich der jeweiligen Betreuung geleistet.

### 4.1 Aufbau und Funktionsweise

Abbildung 5 zeigt graphisch den Aufbau des CIMS mit den oben beschriebenen Komponenten. Im Folgenden wird die Funktionsweise in Worten wiedergegeben.

Zur Startsituation des Systems: Die Servlets werden bei Anforderung vom WWW-Server geladen. Die Servlets haben verschiedene Aufgaben wie die Entgegennahme der Kundenanfragen, die Erfassung von Sekundärinformationen (Web-Seite) und den Aufbau der Expertenstruktur. Gleichzeitig dienen die Servlets als Corba-Clients. Den Corba-Bus initialisieren die Servlets bei ihrer ersten Aktivierung durch den WWW-Server. Der CIMS-Server fungiert auch als Corba-Client. Die Servlets und der CIMS-Server kommunizieren über die ORBs mit den Corba-Servern, die vor der Benutzung gestartet und beim Naming Service angemeldet sein müssen.

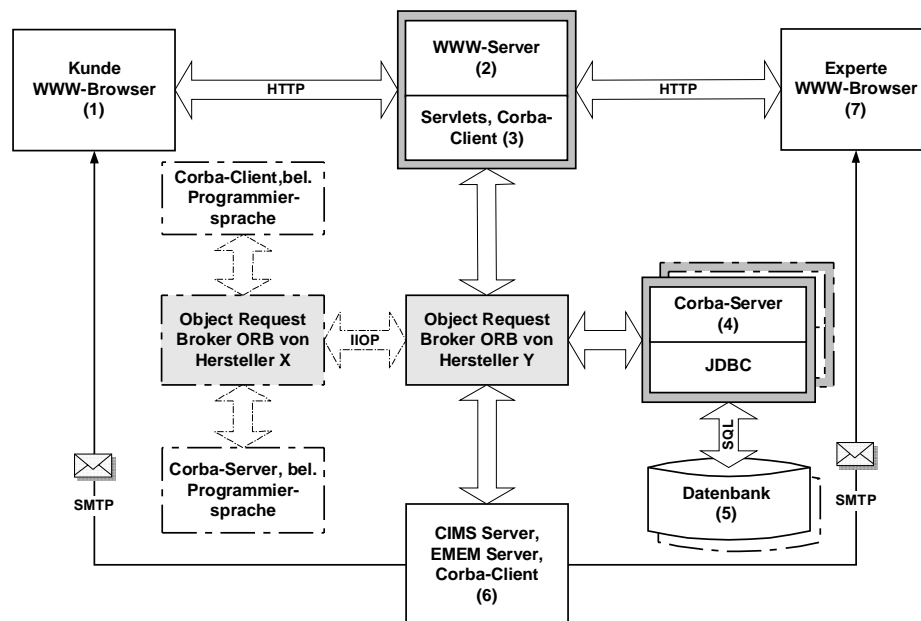


Abb. 5. Prinzipaufbau des CIMS-Prototypen

Die Benutzung des CIMS beginnt damit, daß ein Kunde im Web über seinen Browser (1) auf den WWW-Server (2) gelangt. Falls der Kunde Fragen an das Unternehmen hat, bietet sich ihm die Möglichkeit, über die Web-Seite in Kontakt mit



dem Unternehmen zu treten. Der Kunde bekommt eine HTML-Seite präsentiert, in die er seine Anfrage eingeben kann. Schickt der Kunde das Formular ab, wird im WWW-Server (2) ein entsprechendes Servlet aktiviert, welches die Eingaben des Kunden extrahiert. Darauf holt sich das Servlet in Funktion als Corba-Client (3) über den Nameserver eine Objekt-Referenz für den entsprechenden Corba-Server (4). Über diese Referenz ruft der Corba-Client im Corba-Server Methoden auf. Die von der Anfrage extrahierten Daten dienen der Methode als Übergabeparameter. Die Methoden stellen ihrerseits über JDBC eine Verbindung zur Datenbank (5) her und vollziehen notwendige SQL-Statements.

Bei der Aktivierung des Anfragen-Servlets werden weitere Servlets geladen, die dem Kunden eine Bestätigungsmail schicken und Sekundärinformationen (bspw. Web-Seite) abfragen. Die Sekundärinformationen werden über den gleichen Mechanismus wie die Anfragen des Kunden in die Datenbank eingetragen.

Der CIMS-Server (6) kontaktiert zyklisch über den zugehörigen Corba-Server (4) die Datenbank, um beantwortete Anfragen über E-Mail an den betreffenden Kunden zurückzusenden oder neu eingetroffene Anfragen über die beschriebenen Zuordnungsmechanismen mit Hilfe von E-Mail an den zuständigen Experten (7) weiterzugeben. Die E-Mail für den Experten enthält lediglich einen Link. Über diesen Link erhält der Experte eine dynamisch erstellte HTML-Seite, auf der er die Anfrage des Kunden und einen Link zur gespeicherten Sekundärinformation findet. Den Aufbau der HTML-Seite und die Beschaffung der Sekundärinformation übernimmt ein Servlet indem es über den Corba-Bus eine Corba-Server-Methode aufruft. Der Mechanismus, die Daten über dynamische HTML-Seiten den Experten zukommen zu lassen, ermöglicht eine datenbankgestützte Interaktion.

Der EMEM (Erinnerung-Mahnung-Eskalation-Management)-Server (6) ist ebenfalls als Corba-Client tätig. Dieses Modul überprüft mit denselben Mechanismen wie der CIMS-Server die noch nicht abschließend bearbeiteten Anfragen hinsichtlich ihres Eingangsdatums und löst ggf. entsprechende Maßnahmen aus. Diese Maßnahmen sollen garantieren, daß die Anfragen der Kunden in einem festgelegten Zeitraum beantwortet werden und der Kunde in Problemfällen über den Bearbeitungsstand informiert wird.

Die gestrichelt gezeichneten Komponenten im Bild stellen die Möglichkeiten dar, die sich durch den Einsatz von Corba ergeben. Eine Möglichkeit ist der Zugriff auf die bereitgestellten Komponenten über ein beliebiges anderes Corba Produkt. Die unterschiedlichen Object Request Broker unterhalten sich über das Internet-Inter-ORB-Protocol. Eine weitere Möglichkeit betrifft die Verteilung der Corba-Server und der Datenbank auf verschiedene Rechnersysteme.

## **4.2 Probleme im Zusammenspiel von Java, Servlets, JDBC und CORBA**

Im Zusammenspiel der Technologien ergaben sich keine schwerwiegenden Probleme. Schwierigkeiten traten bei der Installation und im Betrieb der einzelnen Pakete (JDK, JSDK, JavaIDL) auf, da sie sich zumeist noch im Beta-Stadium befanden und die Dokumentation der einzelnen Module sehr dürftig war.

Als erschwerend beim Entwicklungsprozeß erwies sich das Fehlen einer Debug-Umgebung im Bereich Servlets und Corba.

## 5 Zusammenfassung

JavaIDL hält sich bei der Implementation an den Corba-Standard. Die Dokumentation des JavaIDL ORBs fällt spärlich aus. Die wesentlichen Vorteile des JavaIDL sind die freie Verfügbarkeit und die strikte Einhaltung des Standards.

Während der Entwicklung des CIMS wurden auch Versuche mit dem ORB der Fa. Visigenic unternommen. Dieses Produkt ist zur professionellen Entwicklung durch unterstützende Tools und vorhandene Services (Object Request Debugger, Compiler Java-to-IDL, Java-to-IIOP, bessere Unterstützung der dynamischen Objektfindung) besser geeignet als JavaIDL. Der Nachteil der kommerziellen Produkte liegt in der Einführung proprietärer Eigenschaften. Deren Nutzung ist insoweit kritisch, als daß sich daraus Abhängigkeiten zu diesen Produkten ergeben.

## 6 Ausblick

Im Zuge der Weiterentwicklung des CIMS ist beabsichtigt den Zugriff auf die bisher statisch verfügbaren Objekte dynamisch zu gestalten. Desweiteren besteht die Absicht, den Prototypen, der sich in der Laborumgebung als sehr effizient herausgestellt hat, im Praxiseinsatz zu testen.

## Literatur

- [Bishop et al. 97] Bishop, H., Pavlic, C., Hills, B.: Managing Customers with Next-Generation Software Applications: 1997 Edition. Market Research Reports, Aberdeen Group, 1997.
- [Frese & Noetel 92] Frese, E., Noetel, W.: Kundenorientierung in der Auftragsabwicklung. VDI Verlag, Düsseldorf 1992.
- [Gaitanides et al. 94] Gaitanides, Scholz, Vrohling, Raster: Prozeßmanagement - Konzepte, Umsetzungen und Erfahrungen des Reengineering. Carl Hanser Verlag München 1994.
- [Meyer 97] Meyer, F.: Enhancing Commercial Web Sites by Dynamic User-Interaction-Management. WebNet97, AACE, Toronto, 1997.
- [Stauss & Seidel 95] Stauss, B., Seidel, W.: Beschwerdemanagement, Fehler vermeiden – Leistungen verbessern – Kunden binden. Carl Hanser Verlag, München, Wien 1998
- [Sun] Sun Microsystems INC., <http://java.sun.com/products/jdbc/jdbc.drivers.html>, Types of JDBC-Drivers, 1998.
- [Orfali & Harkey 98] Robert Orfali, Dan Harkey, Client/ Server Programming with Java and Corba 2<sup>nd</sup> ed., Wiley & sons INC., 1998.
- [Vogel & Duddy 97] Andreas Vogel, Keith Duddy, Java Programming with CORBA, Wiley & sons INC., 1997.
- [Reese 97] George Reese, Database Programming with JDBC and JAVA, 1<sup>st</sup> ed., O'Reilly & Associates, INC., 1997.
- [Flanagan 98] David Flanagan, Java in a Nutshell (Java 1.1), O'Reilly & Associates, INC., 2<sup>nd</sup> ed., 1998.