Vorwort

»Writing a book is an adventure. To begin with, it is a toy and an amusement; then it becomes a mistress, and then it becomes a master, and then a tyrant. «
–Winston Churchill über seine Memoiren.

Die Welt der Softwareentwicklung ändert sich. Einerseits weitet sich die Durchdringung von **rechnergestützten Systemen** in zunehmendem Maße aus – selbst in Chipkarten, Waschmaschinen und Toastern ist inzwischen eine CPU. Andererseits nimmt die **Vernetzung von Rechnern**, nicht nur, aber in besonderem Maße durch das Internet zu. Dadurch erhält die Entwicklung von **Anwendungen in verteilten Systemen** eine neue Bedeutung.

Verteile Systeme sind in der Industrie und in der Wirtschaft schon lange im Einsatz. Doch während diese Systeme früher für hochspezialisierte Anwendungen entwickelt wurden und ein sehr hoher Aufwand für die Erstellung gerechtfertigt werden mußte, entwickeln sich verteilte Systeme heutzutage langsam zu einer allgegenwärtigen Infrastruktur, die in sehr vielen Bereichen Einzug hält. Während früher nur große Unternehmen, wie Banken und Luftfahrtunternehmen, in der Lage waren, diesen hohen Aufwand zu betreiben, stehen heute der breiten Masse der Firmen und sogar Privatleuten die Möglichkeiten zur Verfügung, verteilte Anwendungen zu entwickeln.

Mit der zum Teil seit Jahrzehnten bewährten Internettechnologie, die durch das WWW populär, weitverbreitet und einfach zu benutzen geworden ist und durch Intra- und Extranets und Electronic Commerce auch eine große wirtschaftliche Bedeutung gewonnen hat, steht eine Technologie und eine Infrastruktur zur Verfügung, die die Entwicklung verteilter Anwendungen ermöglicht.

Der Programmiersprache **Java** kommt in diesem Zusammenhang eine große Bedeutung zu. Einerseits bietet Java die Möglichkeit, Anwendungen durch seine Portabilität und die »Download«-barkeit zu verteilen, d.h., Java-Code kann als Applet von einem Server über das Internet in einen Browser geladen und dort ausgeführt werden. Andererseits bietet sie die Möglichkeit, verteilte Anwendungen zu ent-

wickeln, bei denen die Applikation nicht nur auf einem Rechner, sondern verteilt über mehrere abläuft.

Durch die Kombination dieser beiden Aspekte stehen in Java Möglichkeiten zur Verfügung, die es bisher in keiner anderen weitverbreiteten Programmiersprache gegeben hat. Der erste Aspekt, das Herunterladen von Java-Code als Applet, wurde bereits in vielen Büchern beschrieben. Doch die **Techniken zur Erstellung verteilter Anwendungen mit Java** wurden bisher nur in einzelnen Aspekten und fast ausschließlich in der englischen Literatur behandelt. Dieses Buch stellt in einem Überblick die vorhandenen Techniken zur Entwicklung verteilter Anwendungen mit Java vor.

Aufbau des Buches

Nach einer **Einleitung**, in der die Bedeutung von Java und von verteilten Systemen besprochen und deren Zusammenhänge erläutert werden, gliedert sich dieses Buch in zwei Teile.

- Teil I, **Java in verteilten Systemen**, gibt eine breite Übersicht über die heutzutage verfügbaren Techniken, die in verteilten Systemen Verwendung finden. Dabei wird ein Schwerpunkt auf die drei Aspekte Nebenläufigkeit, Verteilung und Persistenz gelegt, die in fast jedem verteilten System zum Tragen kommen. Die Problematik der Nebenläufigkeit wird mit Java Threads behandelt. Die vorgestellten Techniken zur verteilten Kommunikation reichen von Sockets über RMI und CORBA bis hin zu mobilen Agenten. Die persistente Speicherung wird mit relationalen ebenso wie mit objektorientierten Datenbanken und mit einer Java-Sprachumgebung, die Persistenz unmittelbar in die Sprache integriert, diskutiert. Alle drei Aspekte werden von sogenannten Tupelräumen, einem verteilten Speichermechanismus für nebenläufige Systeme, abgedeckt. Und schließlich wird Jini, eine Infrastruktur, die das einfache Finden von Diensten in verteilten Systemen erlaubt, vorgestellt.

- In Teil II, **Ein verteiltes Java**, wird untersucht, ob sich die Programmierung verteilter Systeme durch eine geeignete Weiterentwicklung der Sprache Java vereinfachen läßt. Es werden zunächst andere verteilte Programmiersprachen untersucht und daraus Grenzen und Möglichkeiten eines verteilten Javas hergeleitet. Danach wird ein Konzept diskutiert, das die Programmierung verteilter Systeme erheblich vereichfacht. Anschließend wird die Programmiersprache »Dejay« – ein verteiltes Java – vorgestellt, die auf diesem Konzept beruht, und mit Beispielen demonstriert.

Die einzelnen Techniken werden mit praktisch relevanten aber einfachen Beispielen vorgestellt. Um eine möglichst gute Vergleichbarkeit verwandter Techniken zu erreichen, wird mehrfach dieselbe Problemstellung in verschiedenen Kapiteln mit unterschiedlichen Techniken gelöst. Die dabei unveränderten Programmteile werden jeweils im Anhang aufgeführt.

Zielgruppe

Dieses Buch ist aus mehreren Lehrveranstaltungen zum Thema Verteilung, Internet und Java an der Universität Hamburg sowie der Forschungsarbeit des Autors auf dem Gebiet Programmiersprachen für verteilte Systeme entstanden. Es richtet sich daher an verschiedene Leserkreise:

- Studenten mit Vorkenntnissen in Java, die sich für die Programmierung verteilter Systeme interessieren. Das Buch kann sehr gut als begleitendes Material für eine Vorlesung, ein Praktikum oder ein Seminar genutzt werden.
- Programmierer, die sich die verschiedenen Programmiertechniken für verteilte Systeme anschauen möchten. Das Buch präsentiert den State-of-the-Art in der Programmierung verteilter Systeme und bietet eine gute Vergleichsmöglichkeit.
- Entscheider mit programmiertechnischem Hintergrund, die ein Gefühl für das Spektrum der verschiedenen zur Verfügung stehenden Alternativen beim entwurf eines verteilten Systems bekommen möchten.
- Forscher, die sich mit dem Stand der Technik vertraut machen und neue innovative Techniken kennenlernen wollen.

Zu allen vorgestellten Techniken werden die wichtigsten Konzepte aufgeführt und an einfachen Beispielen verdeutlicht. Es wird aber in keinem Fall eine Programmierschnittstelle in ihrer vollen Tiefe und Breite abgedeckt. Dieses Buch ersetzt kein Referenzhandbuch. Vielmehr bietet es, was Referenzhandbücher nicht bieten können: einen breiten Überblick über ein Themengebiet in angemessener Tiefe.

Konventionen

Dies ist ein Buch über Java, daher taucht natürlich auch sehr viel Java-Code auf. Die meisten Beispiele sind möglichst kurz gehalten, um das Wesentliche zu verdeutlichen und nicht in einem Wust von

Code untergehen zu lassen. Daher haben die meisten Beispiele keine grafischen Oberflächen, sondern arbeiten mit der Textkonsole als Ein- und Ausgabemedium.

Darstellung von Code

Java-Code und -Bezeichner, wie etwa Klassen- oder Methodennamen innerhalb des Textes, wenn zum Beispiel von einer TestClass die Rede ist, oder Quellcodeauszüge und Bildschirmein- und -ausgaben, werden mit dem Font Sanserif dargestellt. Methoden werden im Fließtext mit leeren Klammern (someMethod()) aufgeführt, unabhängig davon, ob sie Parameter erfordern. Wenn sich der Text auf gewisse Bereiche innerhalb eines Quellcodeauszuges bezieht oder ein bereits zitierter Quellcodeauszug um neue Zeilen erweitert wird, so wird dieser zusätzlich **fett** gedruckt.

```
class TestClass {
  public static void main (String[] args) {
    System.out.println("Hello World");
  }
}
```

Codesegmente, deren Breite über den normalen Textfluß hinausreicht oder die über eine Seite hinausgehen, werden in zwei horizontale Striche gefaßt.

Darstellung von Ein- und Ausgaben

Eingaben, wie zum Beispiel Aufrufe des Java-Compilers javac, werden durch ein vorangestelltes Größerzeichen gekennzeichnet.

```
> javac TestClass.java
```

Ausgaben des Systems auf den Bildschirm, die häufig aus Eingaben resultieren, werden durch einen vorangestellten senkrechten Strich gekennzeichnet.

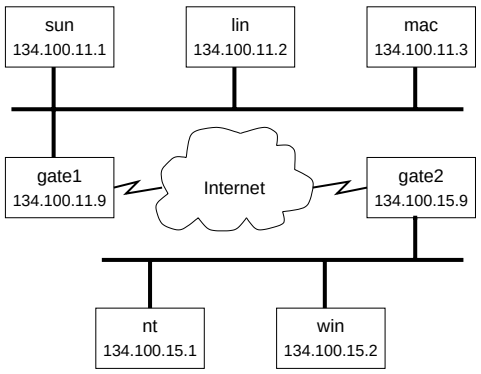

Abbildung 1
Die in Beispielen verwendeten Rechner und die Netzstruktur

```
> java TestClass
| Hello World
```

Da es aber in diesem Buch um verteilte Systeme geht, werden für viele Beispiele mehrere Rechner zur Ausführung einer Anwendung benötigt. Es wird daher hier ein beispielhaftes heterogenes Netz, bestehend aus zwei Teilnetzen verbunden über das Internet, mit verschiedenen Rechnern aufgezeichnet. In diesem Beispielnetz, dargestellt in Abbildung 1, haben die Rechner eine (fiktive) IP-Adresse und einen Namen.

Diese Namen stehen symbolisch für verschiedene Hardwareplattformen und die dazugehörigen Betriebssysteme, die tatsächlich alle als Testrechner für die Programmbeispiele zur Verfügung standen. Die verwendeten Namen sind sun, lin, mac, nt und win sowie gate1 und gate2 als Gateways in das Internet. Wo dies zur Unterscheidung notwendig ist, werden diese Namen oder diese IP-Adressen im Text, in den Programmbeispielen und vor der Eingabe bzw. Ausgabe verwendet. Das folgende Beispiel, in dem eine Nachricht vom Rechner sun an den Rechner lin geschickt und ausgegeben wird, soll dies verdeutlichen:

```
sun> java Talker
sun| sending Message "Hello World" to 134.100.11.2

lin> java Listener
lin| received Message "Hello World" from 134.100.11.1
```

Homepage und Sourcecode

Wie es sich für ein Buch, in dem das Internet eine große Rolle spielt, gehört, gibt es eine Homepage zu diesem Buch, die unter der URL http://www.dpunkt.de/produkte/dejay erreichbar ist. Dort sind Auszüge dieses Buches, die vermutlich unvermeidlichen Errata sowie der Sourcecode zu den Beispielen zugreifbar.

Zu Dejay (ein verteiltes Java, das im zweiten Teil des Buches vorgestellt wird) gibt es eine eigene Domäne. Unter http://www.dejay.org sind Informationen, weitere Dokumentation und der Sourcecode frei erhältlich.

Danksagung

Dieses Buch ist vollständig mit frei verfügbarer Software entstanden. Als Betriebssystem wurde Linux mit der neu entwickelten grafischen Oberfläche KDE eingesetzt. Als Textverarbeitungsprogramm

war kLyX, ein Programm, das die vollen Möglichkeiten von LaTeX ausschöpft, aber seine Komplexität hinter einer sehr gut bedienbaren und hilfreichen Oberfläche verbirgt, von großem Wert. Die Grafiken wurden mit xfig erstellt. Die Programme entstanden im wesentlichen mit dem JDK von Sun in der Version 1.1.7 und sind durchgehend kompatibel zum JDK 1.2 (bzw. der Java-Plattform 2). Für einige Kapitel (Jini und Teile von Tupelräume) ist das JDK 1.2 Voraussetzung. All den Menschen, die an der Entwicklung dieser Software beteiligt waren und ihr Wissen und ihre Arbeitskraft in so selbstloser Weise der Allgemeinheit zur Verfügung stellen, bin ich zutiefst dankbar und möchte ihnen hier eine Würdigung ihrer Arbeit aussprechen.

Dieses Buch ist zu großen Teilen aus meiner Lehrtätigkeit auf dem Gebiet Verteilte Systeme an der Universität Hamburg entstanden. Besonders erwähnen möchte ich das Praktikum »Internet«, das Seminar »Objekte in verteilten Systemen« und das Projektseminar »Realisierung offener verteilter Softwareanwendungen«. Die vielen Studenten, die daran teilgenommen haben, mich das Lehren gelehrt, meine Vorträge konstruktiv und kritisch kommentiert, mich durch ihre eigenen Beiträge weitergebracht, mich zu neuen Ideen inspiriert und schlechte Ideen identifiziert haben, meine Manuskripte gelesen, Fehler gesucht und gefunden, Schwachstellen angemahnt und Vorschläge zur Verbesserung gemacht haben, all denen bin ich mit großem Dank verbunden. Einen ganz besonderen Dank möchte ich denen aussprechen, die mich durch Studien- und Diplomarbeiten ganz unmittelbar in meiner Arbeit unterstützt haben. Bei der Entwicklung von Dejay waren Jan Raap, Thorsten Sturm, Tobias Baier, Nils Poppendiek und Per Fragemann maßgeblich beteiligt. Besondere Erwähnung verdient Marco Kaiser, der mich in vielerlei Hinsicht tatkräftig und zuverlässig unterstützt hat. Für die kritische Durchsicht des Manuskripts möchte ich mich bei Prof. Dr. Claudia Linnhoff-Popien, Dr. Christian Zeidler, Stefan Middendorf und Uta Arnold bedanken.

Bei meinen Kollegen der Arbeitsgruppe Verteilte Systeme möchte ich mich für eine sehr gute Atmosphäre, für viele Gespräche und Diskussionen, für Beistand und Hilfe bedanken. Hochachtung und Dank möchte ich Prof. Dr. Winfried Lamersdorf aussprechen. Er hat mit hohem Engagement und Weitsicht meine Arbeit ermöglicht, begleitet und gefördert.

Schließlich möchte ich dem Team des dpunkt.verlages und insbesondere Christa Preisendanz für die gute Zusammenarbeit bei der Erstellung dieses Buches danken.

Hamburg, im August 1999, Marko Boger

Inhaltsverzeichnis