Zeitschrift: Comtec : Informations- und Telekommunikationstechnologie = information and telecommunication technology

Herausgeber: Swisscom

Band: 78 (2000)

Heft: 5

Artikel: Neue Einsatzgebiete für Java

Autor: Studer, Bruno

DOI: https://doi.org/10.5169/seals-876440

Nutzungsbedingungen

Die ETH-Bibliothek ist die Anbieterin der digitalisierten Zeitschriften auf E-Periodica. Sie besitzt keine Urheberrechte an den Zeitschriften und ist nicht verantwortlich für deren Inhalte. Die Rechte liegen in der Regel bei den Herausgebern beziehungsweise den externen Rechteinhabern. Das Veröffentlichen von Bildern in Print- und Online-Publikationen sowie auf Social Media-Kanälen oder Webseiten ist nur mit vorheriger Genehmigung der Rechteinhaber erlaubt. Mehr erfahren

Conditions d'utilisation

L'ETH Library est le fournisseur des revues numérisées. Elle ne détient aucun droit d'auteur sur les revues et n'est pas responsable de leur contenu. En règle générale, les droits sont détenus par les éditeurs ou les détenteurs de droits externes. La reproduction d'images dans des publications imprimées ou en ligne ainsi que sur des canaux de médias sociaux ou des sites web n'est autorisée qu'avec l'accord préalable des détenteurs des droits. En savoir plus

Terms of use

The ETH Library is the provider of the digitised journals. It does not own any copyrights to the journals and is not responsible for their content. The rights usually lie with the publishers or the external rights holders. Publishing images in print and online publications, as well as on social media channels or websites, is only permitted with the prior consent of the rights holders. Find out more

Download PDF: 29.12.2025

ETH-Bibliothek Zürich, E-Periodica, https://www.e-periodica.ch

Programmiersprachen

Neue Einsatzgebiete für Java

Java ist eine der wichtigsten Programmiersprachen geworden, die sich hervorragend für die plattformübergreifende Kommunikation eignet und immer mehr auch in der Telekommunikation eingesetzt wird.

Im folgenden Artikel werden zwei neue Einsatzgebiete von Java in der Telekommunikation vorgestellt: JAIN (Java APIs for Integrated Networks) und JMX (Java Management Extensions). Hauptziele beider Ansätze sind Dienste-portabilität und Dienstekonvergenz.

BRUNO STUDER, CHUR

Vorteile von Java und Konvergenz

In den letzten Jahren hat sich Java zu einer der wichtigsten Programmiersprachen entwickelt. Auch in der Telekommunikationsbranche bieten die hohe Plattformunabhängigkeit, die Integration als Applets ins Web sowie die objektorientierten Eigenschaften von Java wesentliche Vorteile, sodass Java in der Telekommunikation immer mehr verbreitet sein wird. Während heute noch eher die grafischen Benutzeroberflächen für Telekommunikationssysteme im Vordergrund stehen, wird sich Java in naher Zukunft als Netzwerksprache in der Telekommunikation weiter etablieren und vor allem in Endbenutzer-, Steuer- und Kontroll- sowie Netzwerkmanagement-Applikationen [1] und offenen Service-Creation-Umgebungen verwendet werden [2]. Die Konvergenz zwischen Fest- und Mobilfunknetzen sowie dem Internet schreitet unaufhaltsam voran (Bild 1). Diese Fixed Mobile Internet Convergence (FMIC) wird in [3] genauer beschrieben. Insbesondere die Dienstekonvergenz beinhaltet ein viel versprechendes innovatives Potenzial. Gerade da kommt aber die hohe Plattformunabhängigkeit von Java voll zum Tragen. Mit neuen standardisierten Application Programming Interfaces (APIs) wird es möglich sein, Dienste zu entwickeln, die relativ einfach sowohl in Fest- und Mobilfunknetzen sowie im Internet eingesetzt werden können. Ebenso verhält es sich mit Managementfunktionen. Mit neuen Java-Management-APIs ist ein integriertes Management von Fest- und Mobilfunknetz sowie Internetkomponenten relativ einfach und skalierbar möglich. Im Folgenden werden zwei neue Java-Anwendungsmöglichkeiten in der Telekommunikation beschrieben: JAIN (Java APIs for Integrated Networks) und JMX (Java Management Extensions).

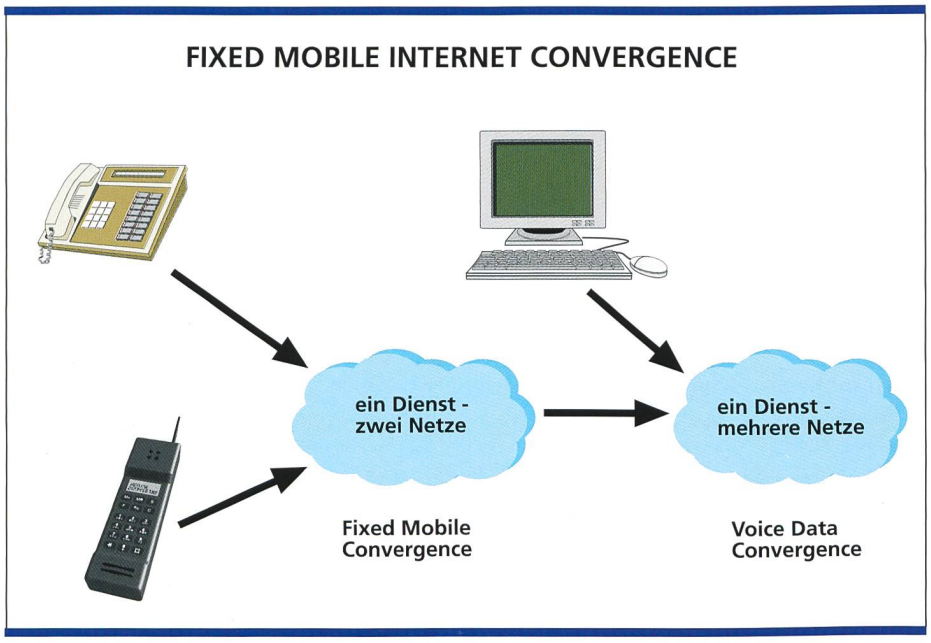

Bild 1. Fixed Mobile Internet Convergence (FMIC).

Netz- und Dienstekonvergenz mit Java

JAIN besteht aus einer Anzahl Network APIs für Java-Plattformen. Mit JAIN ist es möglich, Dienste zu entwickeln, die sowohl für paketvermittelnde Netze (z.B. IP-Netze), leitungsvermittelnde Netze (z.B. PSTN) und Mobilfunknetze (z.B. GSM-Netze) verwendet werden können (Bild 2). Das Hauptziel von JAIN besteht darin, ein Framework für hochportable und konvergente Dienste zur Verfügung zu stellen. JAIN soll eine ganze Reihe von APIs umfassen, welche die Netzwerk- und Protokollimplementationsdetails verbergen und die Realisierung von hochportablen Anwendungen ermöglichen. Im JAIN Protocol Layer werden APIs von der Protocols Expert Group (PEG) spezifiziert, um Schnittstellen für Fest- und Mobilfunknetze sowie dem Internet zur Verfügung zu stellen. Das JAIN SS7 API beispielsweise definiert Java-basierte APIs für die folgenden Protokolle:

- Transaction Capabilities Application Part (TCAP): Das JAIN TCAP API spezifiziert Java-Interfaces und -Klassen, die eine TCAP-Transaktion initialisieren und terminieren. Das JAIN TCAP API wird auf dem ITU-Standard Q.771–Q.775 [4] basieren. Das JAIN TCAP API ist als Java Package jain.protocol.ss7.tcap definiert. Die Spezifikationen liegen seit Ende 1999 zum öffentlichen Kommentieren vor.
- ISDN User Part (ISUP): Das JAIN ISUP API liefert ein API für Signalisierfunktionen, die für Sprach- und Datenkommunikation im ISDN notwendig sind und ebenfalls auf SS7 basieren.
- Mobile Application Part (MAP): Das JAIN MAP API unterstützt den Zugriff auf GSM-Netze und IS41-Spezifikationen. Die JAIN IP APIs umfassen Java-basierte APIs für folgende Protokolle:

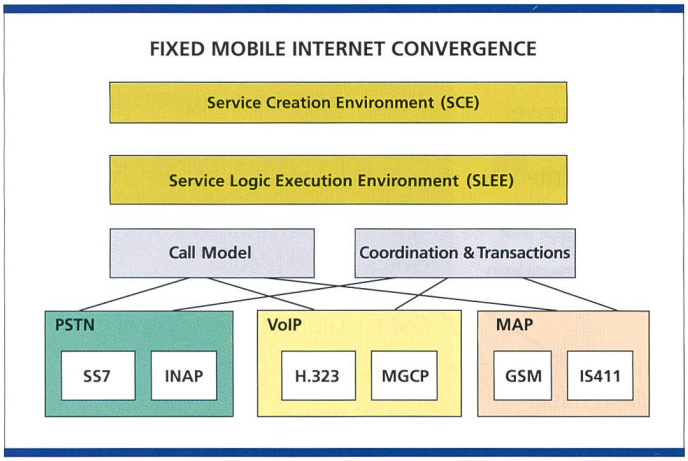

Bild 3. JAIN-Architektur.

- Media Gateway Controller Protocol (MGCP): MGCP kontrolliert Sprach- und Bildkommunikation über paketorientierte Gateways. Die Gateways verbinden das Telefonnetz (PSTN) mit dem Internet. Das entsprechende JAIN MGCP API erlaubt Entwicklern, MGCP-Dienste zu realisieren.
- H.323: Das JAIN H.323 API definiert Echtzeitmultimediakommunikation für paketorientierte Kommunikationsnetze.
- Session Initiation Protocol (SIP): SIP ist ein Client/Server-Protokoll und ermöglicht Anwendern, an Multimediasessionen teilzunehmen. Das JAIN SIP API wird ein generisches, Java-basiertes Interface, das SIP-Kontrollmeldungen zu senden und zu empfangen ermöglicht.

Mit den JAIN SS7 und JAIN IP APIs wird also ein Schwergewicht auf die Konvergenz von intelligenten Netzen (IN) und IP-basierten Netzen gelegt. Deshalb findet man auch häufig für JAIN die Bezeichnung Java Advanced Intelligent Networks. Im Service Layer wird ein JAIN Service Creation Environment (SCE) definiert, bestehend aus Service Building Blocks, die durch grafische Java-Tools oder -Beans repräsentiert werden können. Ein JAIN Call Control (JCC) API und ein JAIN Coordination and Transactions (JCAT) API stellen auf dem Application Layer konsistente Mechanismen für die Verbindungssteuerung zur Verfügung. Bild 3 gibt einen Überblick über die JAIN-Architektur.

Netzwerkmanagement mit Java

Netzwerkmanagement dient dem effektiven und effizienten Betreiben von Netzen aller Art und deckt die funktionalen Bereiche Fault-, Configuration-, Accounting-, Performance- und Securitymanagement (FCAPS) ab. Dazu existieren Lösungen aus der TCP/IP-Welt mit dem Protokoll Simple Network Management (SNMP) und aus der OSI-Welt mit dem Protokoll Common Management Information Protocol (CMIP). Neuere Ansätze basieren auf der Common Object Request Broker Architecture (CORBA). Einen weiteren neuen und vielversprechenden Ansatz stellt die Java-Management-Extensions(JMX)-Architektur dar, auf die im Weiteren genauer eingegangen wird. In allen Ansätzen kommen Managerprozesse und Agentenprozesse vor. Die Managerprozesse sammeln die Managementinformationen der Agenten und stellen sie dem Netzwerkmanager dar. So gibt es auch in der JMX-Architektur JMX-Manager und JMX-Agenten. Die JMX-Manager und -Agenten liefern jedoch eine flexible, verteilte und dynamisch erweiterbare Managementinfrastruktur, realisiert mit Java. JMX-Lösungen sind plattform-, protokoll- und informationsmodellunabhängig. Das heisst, bestehende Managementlösungen lassen sich einfach in JMX integrieren.

Veränderbare Managementdienste

Die konkrete Realisierung von JMX wird von Sun seit Dezember 1999 im Java Dynamic Management Kit (JDMK) 4.0 geliefert [5]. JDMK öffnet die Türe für neuartige, effiziente und flexible Managementapplikationen, indem Managementdienste in autonome Agenten gepackt werden. Diese Agenten sind in der Lage, Managementprobleme selbstständig zu lösen und miteinander zu kommunizieren. Dadurch sind die bis jetzt verwendeten mächtigen zentralen Managementsysteme nicht mehr unbedingt nötig. Die Integration der Managementfunktionen in Agenten bringt folgende Vorteile:

- Reduktion des Managementverkehrs auf dem Netzwerk;
- Low-Level-Probleme können vor Ort behandelt werden, ohne dass Alarme generiert werden müssen;
- schnellere Bearbeitung der (Fehler-) Ereignisse;
- Reduktion der Managementkosten.

JDMK beinhaltet auch einen SNMP-Toolkit, der die Integration von bereits existierenden Standard- oder privaten

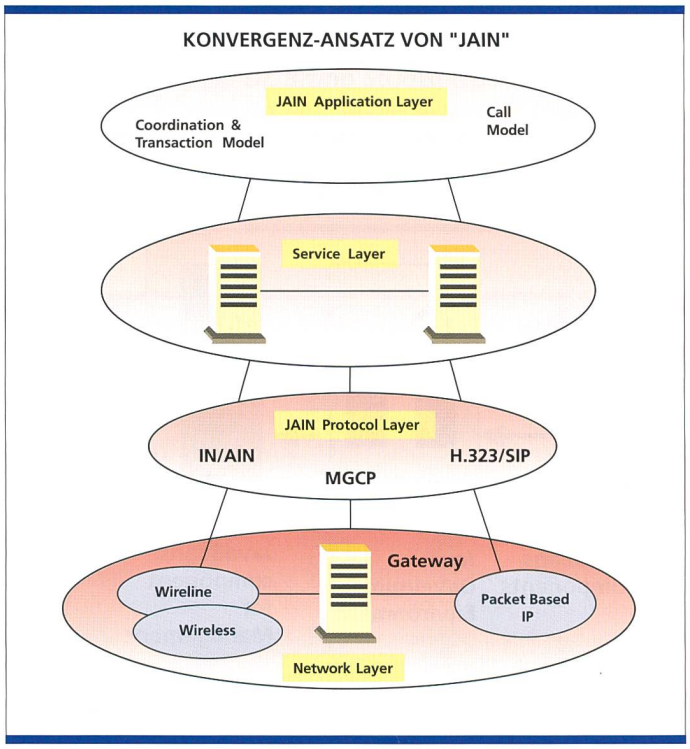

Bild 2. Konvergenzansatz von JAIN.

Links
JAIN: www.sun.com/products/jain
JMX: www.sun.com/products/JavaManagement

MIBs erlaubt. Damit ist es möglich, die heute beliebten und weit verbreiteten SNMP-basierten Managementprodukte einfach zu integrieren. JDMK beinhaltet im Weiteren eine Bibliothek von Managementdiensten, die als Java-Beans-Komponenten zur Verfügung stehen (Bild 4).

Es ist auch möglich, eigene Managementdienste zu entwickeln. Diese können dynamisch im Netzwerk verteilt werden. Deshalb steht ja auch der Name Java Dynamic Management Kit. Im Gegensatz zu traditionellen statischen Managementdiensten können die Managementdienste des JDMK jederzeit mittels des Push-Mechanismus von JDMK aktualisiert und verändert werden. Managementdienste müssen aber nicht ausschliesslich in den Netzwerkkomponenten implementiert sein. Sie können auch netzwerkbasiert in einfachen Webseiten gespeichert werden (ähnlich wie Applets). Zur Boot-Zeit lädt das Java Dynamic Management die neuen Managementdienste vom Webserver. Das bedeutet, dass mittels dieses Pull-Mechanismus in Zukunft nicht schon von Beginn weg feststehen muss, welche Netzwerkkomponenten wie konfiguriert und verwaltet werden müssen, sondern dass Managementdienste je nach Bedarf kreiert und vom Webserver heruntergeladen werden können.

Push- und Pull-Mechanismen des JDMK

Die JDMK-Push/Pull-Software-Verteilungsdienste bieten die folgenden zwei Vorteile:

- Remote Enhancements der Agenten und Manager, die jederzeit über das Web aktualisiert werden können;
- Management von grossen Netzwerken mit minimalem Administrationsaufwand.

JDMK ermöglicht, Netzwerkkomponenten oder -anwendungen, direkt mit einem Webbrowser – mit Java-basierten Applikationen oder mit bereits existierenden SNMP-Managementapplikationen wie Solstice Enterprise Manager oder HP Openview Node Manager – zu verwalten. Eine umfangreiche Anzahl von Protokolladaptoren (HTTP/TCP, HTTP/SSL, RMI, IIOP, SNMP) ermöglicht den Entwicklern, die Managementlösungen ihrer lokalen Kommunikationsumgebung anzupassen. Dadurch ergeben sich folgende Vorteile:

- Schnelle Integration: JDMK-Lösungen können schnell in bereits existierende Managementlösungen eingebaut werden.
- Kompatibilität: JDMK ist voll kompatibel zur Java-Beans-Komponentenspezifikation; irgend eine Bean-basierte Entwicklungsumgebung kann für JDMK-Komponenten verwendet werden.
- Code-Wiederverwendung: JDMK-Komponenten werden als Java-Beans-Komponenten geliefert, die sich gut für eine Wiederverwendung eignen.
- Universell einsetzbar: JDMK-Komponenten können für das Netzwerk-,

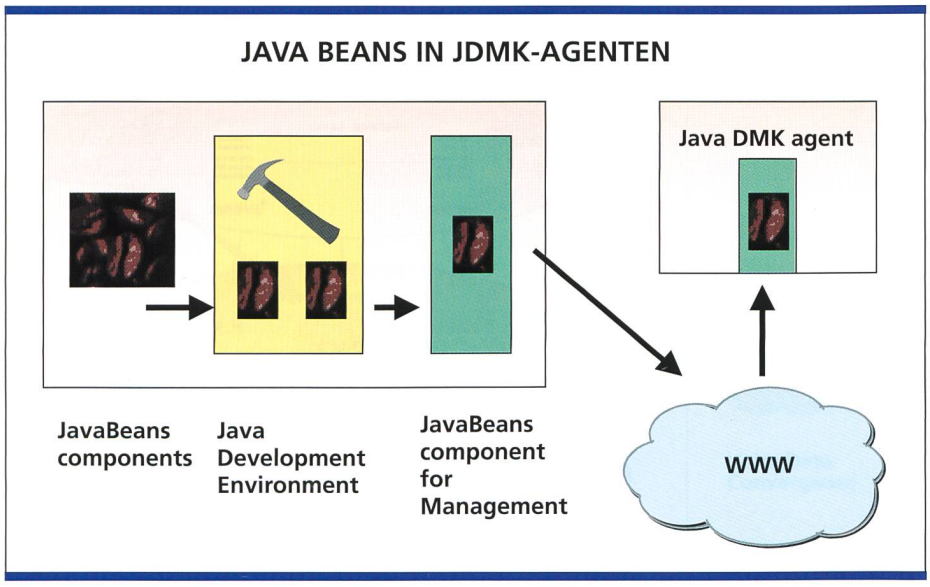

Bild 4. Java-Beans in JDMK-Agenten.

Glossar

AEG	Application Expert Group
API	Application Programming Interface
CMIP	Common Management Information Protocol
FMIC	Fixed Mobile Internet Convergence
GSM	Global System Mobile
IIOP	Internet Inter-ORB Protocol
IP	Internet Protocol
ISUP	ISDN User Part
JAIN	Java APIs for Integrated Networks
JDMK	Java Dynamic Management Kit
JMX	Java Management Extensions
MAP	Mobile Application Part
MIB	Management Information Base
PEG	Protocol Expert Group
PSTN	Public Switched Telephone Network
SNMP	Simple Network Management Protocol
SS7	Signalisiersystem Nr. 7
SSL	Secure Socket Layer Protocol
TCAP	Transaction Capabilities Application Part

Literatur

[1] M. Rubitschun, B. Studer, Java-based Network Management, Singapore International Conference On Networks SICON '98, IEEE Conference Proceedings, World Scientific, Singapore, 1998.
[2] J. de Keizer, D. Tait, R. Goedman, JAIN: A New Approach to Services in Communication Networks, IEEE Communications Magazine, January 2000.
[3] B. Studer, Mit dem Handy ins Internet – Wie Festnetze, Mobilfunknetze und das Internet zusammenwachsen, SEV/VSE Bulletin, 25. Dezember 1999.
[4] ITU-T Rec. Q.771-Q.775, Transaction Capabilities Application Part (TCAP), June 1997.
[5] Java Management Extensions White Paper, Dynamic Management for the Service Age, 1. Juni 1999.

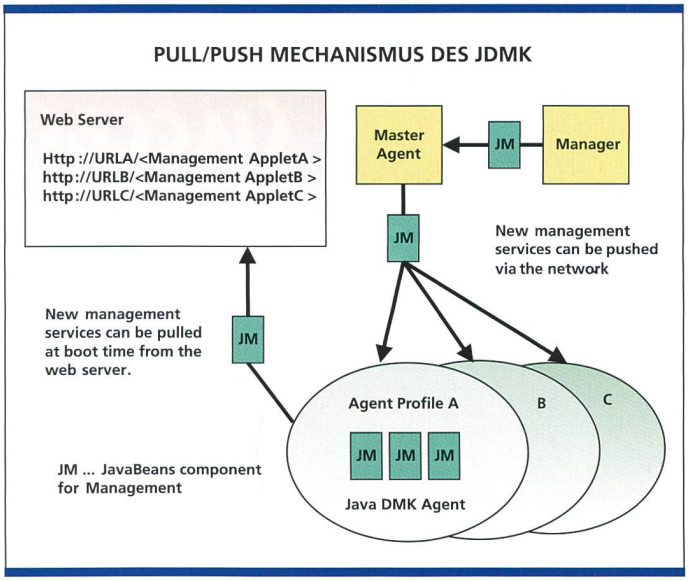

Bild 5. Pull- und Push-Mechanismen des JD.

System- oder Anwendungsmanagement verwendet werden.
- Dynamisch skalierbar: Java Dynamic Management Agents können überall dort eingesetzt werden, wo eine Java-Virtual-Maschine läuft (vom Handy bis zum High End Server). Die Grösse der Agenten hängt nur von den notwendigen Diensten ab (Bild 5).

JDMK umfasst folgende Bausteine:
- eine Java-Dynamic-Management-Runtime-Umgebung;
- eine Bibliothek für wieder verwendbare Managementdienste in der Form von Java-Beans (für z.B. Object Repository, Filtering, Monitoring, Authentication, Discovery);
- ein Managed Object Generator (MOGEN), der die Entwickler für die Erstellung von Clientapplikationen unterstützt;
- ein Java SNMP MIB Compiler, der SNMP MIBs in Java Management Beans übersetzt und damit ermöglicht, dass Java-Dynamic-Management-Agenten von einem SNMP-Manager und umgekehrt JDMK-Manager die SNMP-Agenten verwalten können.

Fazit

Mit JAIN und JMX werden zwei viel versprechende Java Extensions zur Verfügung stehen. Während JAIN erst in der Spezifikationsphase steckt und ein Grossteil der Spezifikationen im Verlaufe des Jahres 2000 publiziert werden sollten, liegt von JMX in Form des JDMK eine konkrete Implementation vor.
JAIN wird die Entwicklung von hochportablen Telekommunikationsdiensten ermöglichen und mit einer Vielzahl von APIs die gängigsten Telefonie-, inklusive der Signalisierprokolle, und selbstverständlich auch IP-basierte Protokolle unterstützen. JAIN ermöglicht somit die Integration von Internet (IP) und intelligenten Netzen (IN) und erlaubt schnell und relativ einfach, neue Dienste zu entwickeln, die sowohl in Fest- und Mobilfunknetzen, aber auch im Internet angeboten werden können. Dies bedeutet eine hohe Diensteportabilität, getreu dem Java-Motto «Write once – run anywhere».
Die JMX-Architektur liefert einen eleganten Java-basierten Managementansatz, mit dem es möglich ist, autonome Managementdienste zu realisieren und über das World Wide Web zu verteilen. Java Dynamic Management Kit (JDMK) ist das erste Java Agent Development Tool auf dem Markt mit Web-basiertem Management und ermöglicht die Integration bereits bestehender Managementprotokolle, wie beispielsweise SNMP. 4.7

Hochschule für Technik und Architektur, HTA Chur

Die HTA (Hochschule für Technik und Architektur) + das IFTW (Institut für angewandte Forschung, Technologietransfer und Weiterbildung) + das Telecomlabor von Swisscom AG in Chur = die TK-Spezialisten in der Schweiz.
Die HTA-Ausbildung ist im TK-Bereich der erste Vollzeitstudiengang in der Schweiz und der einzige deutschsprachige in der Schweiz. Er wird seit über zehn Jahren erfolgreich angeboten. Am IFTW wird ein Nachdiplomstudium NDS Telecom angeboten, das einzige deutschsprachige Hochschulangebot in der Schweiz. Der HTA angeschlossen ist das Telecom Testlabor von Swisscom AG, das einzige deutschsprachige an einer Hochschule. In Chur befindet sich auch das EMV-Prüflabor, neu auch für Mobilfunknetze. Es werden Fachgutachten und Beratungen sowie spezifische Kurse für Fach- und Führungskräfte angeboten.

Info: HTA Chur, Ringstrasse, CH-7004 Chur, Tel. 081 286 24 24, Fax 081 286 24 00, Homepage: www.fh-htachur.ch, telecom.tlab.ch/studer/bruno.htm

Dr. Bruno Studer, *dipl. El.-Ing. ETH, FH-Dozent, Studienleiter NDS Telecom, Hochschule für Technik und Architektur, HTA (Fachhochschule Ostschweiz) Chur*

Summary

Java in telecommunications: JAIN and JMX

Java has become one of the most important programming languages around, being particularly suited to cross-platform communications, and is being used more and more in telecommunications as well. The article introduces two new implementations of Java in telecommunications: JAIN – Java APIs for Integrated Networks and JMX – Java Management Extensions. Both are principally aimed at service convergence and portability.