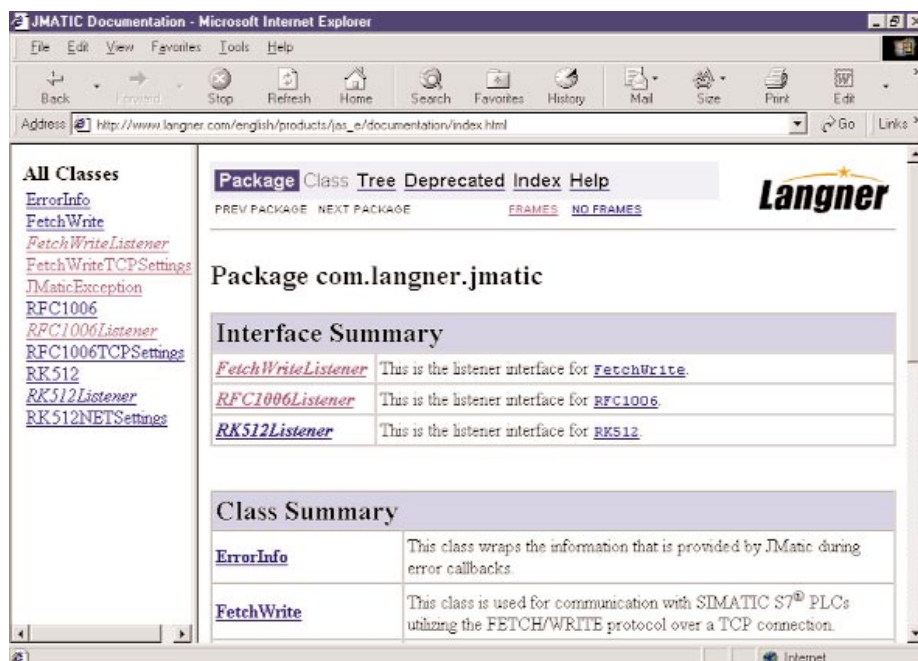


## Java in der Automatisierung:

EXKLUSIV IN  
**elektro**  
AUTOMATION

# Mehr als ein Modethema?

Java gehört zu den Themen, die uns in der Fachpresse und auf Kongressen mit einer gewissen Aufdringlichkeit begegnen. Auf der anderen Seite haben die wenigsten Automatisierungstechniker im realen Leben Projekte gesehen, die auf Java aufsetzen. Anlass genug zu der Frage, ob das Gerede um Java einfach nur Hype ist, oder ob mehr dahinter steckt.



## Gehört bei Java-Produkten immer dazu: HTML-Online-Dokumentation in einem standardisierten Format (JavaDoc)

**U**m es gleich vorweg zu nehmen: Es steckt mehr dahinter. Aber die Vorteile und Einsatzbereiche von Java in der Automatisierung sind begrenzt, und die Einbeziehung von Java in bisherige Architekturen bringt Probleme mit sich. Hier eini-

ge Erfahrungen eines Softwarehauses, das seit über zehn Jahren IT-Projekte in der Automatisierung umsetzt und dabei von Step7 bis Java mit unterschiedlichen Software-Entwicklungsumgebungen gearbeitet hat.

## Java in der Praxis

Wo treffen wir in der Automatisierung überhaupt auf Java? Hauptsächlich dort, wo es um die Verknüpfung von Automatisierungstechnik und IT-Anwendungen geht. Zwar findet sich Java auch in eingebetteten Systemen (embed-

ded Java), in Deutschland haben solche Plattformen bislang allerdings keine signifikante Marktverbreitung. Ein Grund hierfür mag der relativ hohe Speicherbedarf sein: 2 MB ROM und noch mal 2 MB RAM sollten es schon mindestens sein, wenn das eingebettete System Java sprechen soll.

## Vorteil Plattform-unabhängigkeit

Der größte Vorteil von Java und oft auch der Hauptgrund, weshalb Java in Automatisierungsprojekten eingesetzt wird, ist die Plattformunabhängigkeit dieser Sprache. Ein Java-Compiler erzeugt plattformunspezifischen Zwischencode, der von der Java Virtual Machine des Zielsystems interpretiert wird. Somit ist es zum Beispiel möglich (und funktioniert auch tatsächlich in der Praxis), dass die Entwicklung eines Java-Programms auf einem Windows-PC stattfindet, das Zielsystem jedoch Linux als Betriebssystem verwendet.

Zwei aktuelle Kundenprojekte von Langner belegen dies. Ein Ausrüster von Ölbohrplattformen hat sein Leitsystem komplett in Java geschrieben. Grund: die Systemumgebung umfasst Single-Board-PCs mit Hard Hat Linux als Betriebssystem, mobilen Web Pads, Windows-PCs und Unix-Servern. Mit klassischen PC-Programmiersprachen wie Visual Basic ist dieser Vielfalt nicht Herr zu werden, mit Java schon. Denn Java ist die einzige Softwareumgebung, die wirklich auf allen Systemen bis hin zu PDAs und Handies zur Verfügung steht. Der Part von LANGNER bestand darin, ein Kommunikations-Bindeglied zu SIMATIC-S7-Steuerungen zu liefern. Da als Übertragungsmedium zur Steuerung TCP/IP verwendet wird, stellt dies kein großes Problem dar, zumal in Java bereits eine sehr gute TCP/IP-Unterstützung eingebaut ist. – In einem anderen Projekt ging es für einen großen deutschen Motorenbauer darum, das IT-technisch überkommene Leitsystem durch moderne Technologie zu ersetzen, wobei die bestehende Steuerungs-



Ralph Langner  
Vorstand der Langner  
Communications AG  
in Hamburg

architektur (SIMATIC S5) beibehalten werden sollte. Eine nicht ungewöhnliche Situation: IT-Technik altert schneller als Automatisierungstechnik. Da die neue Hardware schwerpunktmäßig auf Sun-SPARC-Servern aufsetzt, fiel die Entscheidung für Java leicht. Das Softwarehaus hat hier das Kommunikations-Bindiglied von Java zur S5 geliefert.

## Vorteil Entwicklungsressourcen

Außer der Plattformunabhängigkeit gibt es noch einen weiteren gravierenden Vorteil, der für Java spricht. Java ist inzwischen eine sehr verbreitete Programmiersprache, verbreiteter als Basic, Delphi oder C. Es ist deshalb leicht, Programmierer zu finden, die Java beherrschen. Weltweit gibt es heute geschätzte 2,5 Millionen Java-Programmierer. Diese Zahl wird voraussichtlich auf 4 Millionen im nächsten Jahr wachsen. Erfreulich ist ebenfalls die große Zahl erschienener Fachliteratur und Veranstaltungen (Seminare, Kongresse usw.), die sich Java widmen.

Schließlich gibt es für Java auch einen Produktmarkt für Komponenten und kompletten Anwendungen. Obwohl im Bereich der SPS-Anbindung das Langner-Produkt JMatic allein auf weiter Flur steht, findet man auf [www.java.sun.com](http://www.java.sun.com) unter dem Stichwort Manufacturing immerhin 74 Produkteinträge, darunter komplette Java-basierte MES-Systeme wie zum Beispiel den "Illuminator" der US-Firma Lighthammer.

## Web-Browser und Web-based Services

Wo Web-Browser zur Visualisierung eingesetzt werden, ist Java nicht weit. Der Grund ist einfach: Java ist bislang die technisch einzig überzeugende Lösung, Programme und Programmschnipsel (Applets) im Browser ablaufen zu lassen. Dieser Bereich wird allerdings häufig überschätzt und besonders anfällig für Hype-Auswüchse. Die Visualisierung per Browser oder der Web-Server im Feldgerät ist keine technologische Innovation, die einen Mehrwert schafft. Der Grund hierfür: Browser werden immer vom Menschen bedient. Produktivitätsfortschritte lassen sich aber vor allem dort erzielen, wo Informationen automatisiert, also ohne

Zutun eines menschlichen Bedieners verarbeitet werden. Web-Browser und ebenso die damit zusammenhängenden Techniken und Protokolle sind dazu ungeeignet. Wir gehen deshalb davon aus, dass Browser-Applets nur ein Zwischenschritt sind in Richtung auf so genannte Web-based Services, bei denen Daten nicht von einem menschlichen Benutzer abgerufen werden, sondern automatisiert durch andere Anwendungen. Auch hierfür ist Java dann allerdings wieder prädestiniert durch seine hervorragende Unterstützung von XML und SOAP (Simple Object Access Protocol). Die in diesem Kontext von Langner entwickelte FactoryXML™-Technologie wird ebenfalls hauptsächlich per Java an Internet-Anwendungen angebunden.

## Der Kulturschock

Soweit die Vorteile. Nachteile gibt es leider auch. Java ist für Automatisierungstechniker alles andere als eine leicht verständliche Sprache. Tatsächlich ist Java so ziemlich das Gegenteil dessen, was SPS-Programmierung ausmacht. Ein SPS-Programmierer, dessen täglich Brot AWL, KOP oder FUP ist, und ein PC-Programmierer, der Web Based Services in Java entwickelt, leben in verschiedenen Welten. Sie denken noch nicht einmal in der gleichen Weise. Sie sind so unterschiedlich wie Rambo und Harry Potter – müssten die in einem Film gemeinsam auftreten, wäre das Happyend nicht garantiert. Das ist also keine gute Voraussetzung für Projekte, die beide Bereiche umfassen. Wir haben diesen Kulturschock in unseren Schulungen, in denen oft ein gemischtes Publikum von SPS-Programmierern und IT-Entwicklern sitzt, häufig beobachtet.

Die wichtigsten Unterschiede zwischen einem SPS-Entwicklungssystem wie zum Beispiel Step7 und Java sind: Erstens, Java ist eine objektorientierte, sehr stark abstrahierende Sprache. SPS-Entwicklungssysteme sind alles andere als objektorientiert. Was der Java-Programmierer produziert, erscheint dem Automatisierer abgehoben; eine SPS-Anweisungsliste mit Befehlen, die aus einem Buchstaben bestehen, erzeugt beim Java-Crack ungläubiges Staunen. Das objektorientierte Denken ist der SPS-Programmierung fremd. Zweitens, SPS-Programme laufen sequenziell ab, Java-Programme dagegen ereignisgesteuert. Die für SPS-Programme typischen Pol-

ling-Schleifen (Programmzyklen) sind in Java der Super-GAU, und manchem Automatisierer erscheint die Java-Bearbeitung von Ereignissen über Listener-Objekte als Teufelszeug. – Diese Unterschiede sind tief in gänzlich verschiedenen Systemarchitekturen begründet und werden auch in Zukunft weiter bestehen. Gefragt sind deshalb Entwickler, die in beiden Bereichen Know-How aufgebaut haben.

## Und was ist mit .NET?

Wir haben uns inzwischen daran gewöhnt: Jeder nennenswerte Erfolg irgendeiner Technologie oder Anwendung, die auch nur entfernt mit Datenverarbeitung zu tun hat, ruft den größten Softwarehersteller der Welt auf den Plan, der sich auch eine Scheibe vom Kuchen abschneiden möchte (und dann wenig später die Produktion eigener, besserer Kuchenformen ankündigt). Selbst wenn es sich um Spielekonsolen handelt: Wo neue Märkte entstehen, mischt Microsoft mit. So war es auch nur eine Frage der Zeit, bis Java von Microsoft mit einer Konkurrenztechnologie namens .NET gekontert wurde. Und obwohl freigegebene .NET-Produkte erst seit einigen Wochen verfügbar sind, haben die Buschtrommeln dafür gesorgt, dass die Microsoft-Technologie bereits in aller Munde ist.

## Die technischen Unterschiede sind gering

Die technischen Unterschiede beider Plattformen sind vergleichsweise gering. Eine Entscheidung für die eine oder die andere Technologie wird sich daher eher an nicht-technischen Kriterien orientieren. .NET ist das Produkt eines einzelnen Herstellers, Java ist ein Standard, auf den sich mehrere große Hersteller geeinigt haben (Sun Microsystems, IBM, BEA, Oracle und andere). Wer seit Jahren schon auf Microsoft-Produkte setzt und daran glaubt, dass die Gates-Company auch .NET im Markt durchdrücken wird, wird sich gegen Java entscheiden. Wer die Vorteile einer ausgereiften Produktinfrastruktur überzeugend findet, setzt auf Java.

Weitere Informationen

► eA 513