

System Services

APL2-Schnittstellen zu System-Services werden durch *synchron arbeitende assoziierte Prozessoren* mit Namensassoziiierung und durch *asynchrone Hilfsprozessoren* bereitgestellt. Diese Hilfsprozessoren laufen in eigenen Adressräumen und kommunizieren mit zugehörigen Anwendungen über den Shared Variable Processor (SVP) und gemeinsame Variable.

Als synchrone assoziierte Prozessoren werden mitgeliefert:

- P10 *REXX-Prozessor*: erlaubt die Benutzung von REXX-Funktionen wie APL2-Funktionen, Manipulation von REXX-Programmen wie APL2-Strukturen und leichten Zugriff auf Betriebssystemdateien.
- P11 *Aufruf externer Routinen*: ein APL2-Programm kann damit Routinen außerhalb des APL2-Arbeitsbereiches aufrufen, die sich in einem APL2-Namespaces oder in der Bibliothek einer anderen Programmiersprache befinden. zahlreiche solcher externer P11-Routinen und Dokumentationen dazu werden mitgeliefert, sowie auch Musterprogramme als Basis für Entwicklung eigener externer Routinen.
- P12 *Dateien als Variable*: Dateien (Textdateien des Betriebssystems und solche im AP210-Format) können wie APL2-Variable mit APL2-Elementarfunktionen und -Operatoren verarbeitet werden.
- P14 Ermöglicht den Zugriff auf Java-Felder und -Methoden. Das APL2 Programming Interface (APL2PI) arbeitet mit P14 und bildet die Brücke zu IBM WebSphere.

Mitgelieferte asynchrone Hilfsprozessoren:

- AP100 *für Host-System-Befehle*: übermittelt Host-System-Befehle aus APL2-Anwendungen
- AP101 *für Alternativ-Eingabe (Stack)*: zur Erstellung von programmierten Alternativ-Eingaben an APL2.
- AP119 *TCP/IP Socket API*: leitet Aufrufe an TCP/IP weiter, das die Netz-Kommunikation sicherstellt.
- AP120 *APL2-Sitzungsmanager*: er ermöglicht dem Benutzer die Kontrolle über sämtliche von ihm gestarteten APL2-Prozesse, selbst auf entfernten APL2-Systemen, auch wenn sie auf einer anderen Plattform laufen.
- AP127 *DB2-Prozessor*: zum Zugriff auf IBM DB2 (auch auf verteilte DB2-Systeme) über die SQL-Sprache.
- AP144 *X Window API (nur für APL2 auf Unix-Plattformen)*: ist die Schnittstelle zu einer großen Zahl von Xwindow-Systemaufrufen (Xlib). Arbeitsbereiche für Cover-Funktionen und Demos sind beigefügt.
- AP145 *GUI Serviceprocessor (nur für APL2 unter Windows)*: ermöglicht die einfache Verwendung zahlreicher Objekte der Windows Services zur grafischen Benutzerführung. Arbeitsbereiche für Cover- und Beispiel-Funktionen sowie Demos sind beigefügt.
- AP207 *Universeller Grafikprozessor*: erzeugt Grafiken, Texte und Bilder von hoher Qualität mit Hilfe einer einfachen Befehlssyntax. Arbeitsbereiche für Demos und GRAPHPAK sind beigefügt.
- AP210 *Dateiprozessor*: zum Schreiben und Lesen von Dateien des Betriebssystems. Diese können Texte, binäre Daten und APL2-formatierte Objekte enthalten.

- AP211 *APL2 Objektbibliotheks-Prozessor*: zum Schreiben und Lesen von APL2-Strukturen über ihre Namen. Die damit erzeugten Dateien sind portabel auf alle APL2-Plattformen (die von APL2/DOS sind lesbar).
- AP227 *ODBC-Prozessor*: zum Zugriff über SQL auf alle Datenbank-Systeme und Anwendungen, die das ODBC-Protokoll unterstützen.
- AP488 *GPIB-Prozessor*: zum Zugriff auf externe Geräte, die über den General Purpose Interface Bus (GPIB) und National Instruments GPIB Treiber angeschlossen sind. (nur für Windows Systeme).

Technische Informationen

Hardware-Voraussetzungen:

- Mindestens 25 MB freier Plattenplatz
- Mindestens 15 MB Hauptspeicher über den Bedarf des Betriebssystems hinaus

Software-Voraussetzungen:

Workstation APL2 Version 2.0 hat die folgenden Software-Mindestvoraussetzungen:

- AIX 5.0 mit Motif 1.2 und X Window System X11R5
- Eine Linux-Distribution für PC-kompatible Maschinen mit Motif 2.1 und X Window System X11R6
- Solaris 2.5 mit Motif 1.3 und X Window System X11R5
- Windows 98, ME, NT, 2000 oder XP

Einige Einrichtungen von Workstation APL2 haben zusätzliche Mindestvoraussetzungen:

- Adobe Acrobat Reader ist erforderlich zur Darstellung der Online-Dokumentationen.
- IBM DB2 Universal Database, Version 5.0 für AP 127
- Ein ODBC Driver Manager, der ODBC 3.0 Application Programming Interfaces unterstützt, für AP 227.
- TCP/IP für AP 119 und systemübergreifende gemeinsame Variable.
- IBM Object REXX zum Aufruf von REXX-Routinen mit Prozessor 10
- TCL und TK 8.1 für die externe TCL-Funktion.
- Eine lauffähige Java-Version für Kompression/Dekompression

Englische Dokumentationen:

Workstation APL2 Version 2.06 enthält die folgenden englischen Dokumentationen im PDF-Format:

- | | |
|---|-----------|
| ➤ APL2 User's Guide | SC18-7021 |
| ➤ APL2 Programming: Language Reference | SH21-1061 |
| ➤ APL2 Programming: Using SQL | SH21-1057 |
| ➤ APL2 Programming: Using APL2 with WebSphere | SH18-9442 |
| ➤ APL2 Programming: Developing GUI Applications (nur für Windows) | SH18-7383 |

- APL2 GRAPHPAK:
User's Guide and Reference SH21-1074
- APL2 Language Summary SX26-3851

Kompatibilität:

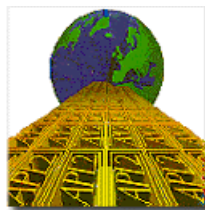
Workstation APL2 V2.0 ist kompatibel mit den APL2V1-Produkten für AIX, OS/2, Sun/Solaris und Windows. Mit diesen Produkten geschriebene Anwendungen laufen auch auf Workstation APL2 V2.0, das aber ein neues Format für die Arbeitsbereiche benutzt. D.h., mit V2.0 gespeicherte Arbeitsbereiche können nicht in V1.0-Systemen geladen, sondern nur durch)OUT und)IN zurück migriert werden .

Hinweis:

Diese Veröffentlichung von DPC ist eine auf deutsche bzw. europäische Verhältnisse angepasste Übertragung des IBM Dokuments „Workstation APL2 for Multiplatforms Version 2.0 Technical Brief“ vom Dezember 2004 und weiterer IBM Veröffentlichungen vom Mai 2005. Sie kann nur allgemeine Hinweise geben. IBM, APL2, OS/2, AIX, und DB2 sind Warenzeichen der IBM Corporation. Sun und Solaris sind Warenzeichen von Sun Microsystems, Inc. Windows, Windows 98, Windows ME, Windows NT, Windows 2000, Windows XP und COM sind Warenzeichen von Microsoft Corporation. Linux ist ein Warenzeichen von Linus Torvalds. Adobe und Acrobat sind Warenzeichen von Adobe Systems, Inc. Motif ist ein Warenzeichen von The Open Group.

i Workstation APL2 Version 2.0 kann bei DPC nicht nur neu bestellt, sondern auch von einem der Vorgängerprodukte APL2 Version 1.0 für AIX, OS/2, Solaris oder Windows aufgerüstet werden, beides über das IBM-Vertriebsprogramm „Passport Advantage“. Weitere Informationen findet man auf den APL-Webseiten von IBM oder von DPC:

- <http://www.ibm.com/software/admttools/apl>
- <http://www.dpc.de>



Kontakt:

Dittrich & Partner Consulting GmbH, Kieler Str.17, D-42697 Solingen, Tel. 0212-26066-0, Fax: -26066-66, eMail: info@dpc.de

IBM „APL Products and Services“: eMail: apl2@vnet.ibm.com, Fax: 001-408-463-4488, Tel. 001-408-463-APL2 (001-408-463-2752)

Der Buchtip

GUI-Utilities für Dyalog APL/W

Ein Buch von Kai Jäger, erschienen im Selbstverlag (BoD), erhältlich im Buchhandel und bei Internet-Anbietern wie Libri und Amazon mit der ISBN-Nr. 3-8311-1831-0.

Das Buch beschreibt frei verfügbare Software, mit deren Hilfe die ohnehin hohe Produktivität eines APL-Programmierers noch beträchtlich erhöht werden kann. Diese Software reduziert den Aufwand für die Erstellung eines GUIs erheblich, wenn die Aufgabenstellungen aus dem kaufmännischen Bereich stammen.

Das Buch wendet sich an den gestandenen APL-Programmierer, der Anwendungen realisiert. Für den APL-Neuling ist es nicht geeignet. Es ist insbesondere keine Einführung in die GUI-Programmierung mit APL. Es ist nur für Dyalog APL/W nutzbar.

Wer GUIs erstellen muss, der wird in der verfügbaren,

außerordentlich mächtigen Software eine große Hilfe finden. Mit Hilfe dieses Buches ist auch die Einarbeitung in die grundlegenden Konzepte kein Problem mehr.

Wer APL als mathematisches Werkzeug für statistische Untersuchungen oder zur Datenanalyse benutzt, wird die hier vorgestellte Software nur dann sinnvoll nutzen können, wenn er zumindest gelegentlich auch GUIs erstellen muss.

Nebenbei lernt man noch frei verfügbare Software kennen, welche die Verwendung von Software-Modulen anderer Autoren in eigenen Applikationen stark vereinfacht. Auch Programmier-Teams werden durch diese Software unterstützt.

Weitere Informationen finden Sie unter:
<http://www.kai-jaeger.de>

Nancy Wheeler

Workstation APL2

Version 2 Service Level 6

In May 2005, Service Level 6 to Workstation APL2 Version 2 was made available. In addition to the usual fixes for reported problems, this service level contains several interesting enhancements that address requirements submitted to IBM by the German customers of APL2.

Compression Utilities

A new set of external functions provides the ability to compress APL2 objects. The resulting compressed objects take up much less space than the objects in their original form, and thus are more economical when sent across networks. There are two pairs of compression functions.

- ZIP and UNZIP operate on individual arrays. The array is passed as an argument to ZIP and the compressed object is returned as its result. To uncompress, pass the compressed object to UNZIP and the original array is returned as the result.
- ZIPWS and UNZIPWS operate on the entire active workspace. ZIPWS compresses each object in the workspace and catenates them into a single compressed object. Once the compressed object is created, the original objects are erased, and)SAVE will produce a much smaller workspace file. After)LOAD, UNZIPWS is used to restore the original objects and erase the compressed object. The argument to both ZIPWS and UNZIPWS is the name to be used for the compressed object.

The compression utility functions are made possible by the use of built-in facilities of Java and the APL2 Java interfaces.

COM Interface for Windows

A new external function provides access to the Microsoft *Component Object Model* (COM) from APL2.

The Microsoft Component Object Model is a system for creating software components that can interact. COM is the foundation technology for Microsoft's OLE (compound documents), ActiveX (Internet-enabled components), and others.

Many applications, such as Microsoft's Office suite of products and elements of Windows, are distributed as COM components. Through the COM external function, these applications can be used from APL2.

COM is an object-oriented system. This means that COM components are distributed as *classes* and operate as *objects*. An application, such as APL2, creates an *instance* of a COM class to create a COM *object*. Once an object is created, the application can use *methods* and *properties* within the object. When the application has finished using the object, it releases the object to free its resources.

The following log demonstrates using COM to control Microsoft Word:

```

3 11 DNA 'COM'
1
  A Start Microsoft Word
WORD+COM 'CREATE' 'Word.Application'

  A Open a document
FILE+COM 'METHOD' WORD 'Documents.Open' 'c:\sample.doc'

  A Get the builtin style constants
CONS+COM 'QUERY' 'CONSTANTS' WORD 'WdBuiltinStyle'

  A Get the value of the constant for level 1 heading style
INDEX+CONS [CONS;1] c'wdStyleHeading1';2]

  A Get the font used for level 1 headings
COM 'PROPERTY' FILE 'Styles.Item().Font.Name' INDEX
Verdana

  A Set the font used for level 1 headings
COM 'PROPERTY' FILE 'Styles.Item().Font.Name' INDEX 'Arial'

  A Save the changes
COM 'METHOD' FILE 'Save'

  A Close the file
COM 'METHOD' FILE 'Close'

  A Release the file
COM 'RELEASE' FILE
  A Close Word
COM 'METHOD' WORD 'Quit'

  A Release Word
COM 'RELEASE' WORD

```

The COM function opens up many new possibilities for combining APL2 with other Windows programs.

■ Calls to APL2 from Visual Basic

Visual Basic has been added to the list of languages and programs that are supported by the APL2 Program Interface (Calls to APL2). A set of subroutines is provided for use as Visual Basic external procedures. These routines allow you to manage an APL2 session from within your Visual Basic program. For example, Excel macros written in Visual Basic can call APL2 to perform calculations.

The Visual Basic to APL2 interface is similar to that already provided for C and Java. Routines are provided to start and end an APL2 session, manage APL2 objects in the workspace, convert APL2 objects to and from Visual Basic *variant* form, and execute APL2 functions and expressions. A complete sample Visual Basic program is shown in the *APL2 User's Guide* along with examples of how to call each routine.

■ Sample C Program for Calls to APL2

While the *APL2 User's Guide* does include program segments that show how to use the APL2 Programming Interface to request the various services supported by the C interface, it does not show a complete, working C program that demonstrates the entire process. Such a program is now being shipped with APL2. On Windows, find it in the APL2 \samples directory. On Unix systems, it is located in the /examples/qna directory.

■ Windows Session Manager

A new option has been added to the Session Manager's *Object Popup Menu* on Windows. If the mouse pointer is positioned over the name of a variable, a right click and selection of the *Display* option will open a window containing the output of the APL2 DISPLAY function for that variable.

■ AP 145 Extensions

Several extensions have been made to AP 145 in this service level:

- The Treeview control is added to the list of Windows Common Controls supported by APL2.
- The new NAME property allows controls within a dialog to be identified by character names rather than numbers.
- The FOCUS property for Listview controls returns the index of the item that currently has focus.
- New events supported are Kill focus and Set focus for Listview controls, Start Menu for dialogs, and Context Menu.
- The Resizable style for Grid controls allows new rows and columns to be added under program control.

Kontakt:

Nancy Wheeler , eMail: nwheeler@us.ibm.com

Die längste elektronische Schultafel Berlins

FU-Informatiker entwickeln intelligenten Hörsaal

Berlin. Der erste elektronische Hörsaal ist an der Freien Universität Berlin eingeweiht worden. Kernstück des Vorlesungssaals sind vier berührungssensitive Projektionsflächen mit einer Gesamtlänge von sechs Metern. Berlinweit handelt es sich hierbei um den größten, berührungsempfindlichen Computerbildschirm. Der Dozent kann an den elektronischen Bildschirmen direkt arbeiten, Worte oder Diagramme wie an einer herkömmlichen Kreidetafel anbringen. Doch die Tinte ist nur noch digital: Statt Kreide benutzt der Dozent einen Laserstift, mit dessen Hilfe das Tafelbild von einem Computer erzeugt wird. Der Vorteil für die Studierenden liegt auf der Hand: Ist die Vorlesung zu Ende, steht die Transkription bereits im Netz. Die Studierenden können sich auf die Vorlesung konzentrieren und müssen nicht mehr mitschreiben. Das Institut für Informatik hat den "intelligenten Hörsaal" vor wenigen Wochen eingeweiht, doch das E-Kreide-Projekt läuft bereits seit fast fünf Jahren und hat europaweit mehrere Preise gewonnen.

Neben der Benutzung als herkömmliche Tafel sind Bilder und interaktive Internet Dienste während des Vortrags in das Tafelbild integrierbar. Diese können direkt von der Festplatte oder aus dem Internet geladen werden. E-Chalk bietet ausserdem Formelwertung und Funktionsplot über eine Handschrifterkennung. Der Zuhörer kann den Vortrag im Hörsaal oder zu Hause mitverfolgen. Hierzu wird nur ein normaler Browser benötigt. E-Kreide überträgt und speichert gleichzeitig Tafelbild, Audio und Video. Vorlesungen werden so gespeichert, daß sie jederzeit aus dem Internet abspielbar sind. Von dem Tafelbild wird außerdem ein PDF erzeugt, so daß die Zuhörer in der Veranstaltung nicht mehr mitschreiben müssen.



Moderne E-Tafel: Statt Kreide benutzt der Dozent einen Laserstift
(Foto: FU Berlin)

Die intelligente Kreide-Tafel kann die Dozenten bei verschiedenen Aufgaben unterstützen: Die Tafel übersetzt Sprachen, kann mathematische Funktionen als Bilder wiedergeben oder logische Schaltungen erkennen und zum Laufen bringen. Auch Computerprogramme können handschriftlich eingegeben werden und sind sofort lauffähig.

Rojas schreibt in grüner Farbe die mathematische Formel $f(x,y) = xy$ an die E-Tafel und während er noch erklärt, baut sich auf der Plasmaoberfläche langsam der zugehörige dreidimensionale Graph auf. "Hierzu benutzen wir einen kommerziellen algebraischen Server", sagt der Informatik-Professor. Wer wissen möchte, was das Wort "Kreide" auf Englisch heißt, braucht es ebenfalls nur auf die Tafel schreiben und schon erscheint der englische Begriff "chalk" - allerdings ebenfalls nur, wenn über einen Script eine Verbindung zum Internet besteht und der Dozent zuvor ein "bookmark" zu der entsprechenden Seite gesetzt hat.

Kontakt: Prof. Dr. Raúl Rojas, Institut für Informatik der Freien Universität Berlin, Projekt E-Chalk, Takustr. 9, D-14195 Berlin, Tel. 030.838-75130, eMail: rojas@inf.fu-berlin.de, Internet: www.e-creide.de, www.inf.fu-berlin.de/.

Axel Holzmüller

APL2 und Java

vorwärts und rückwärts

■ Rückblende: Java

- objektorientierte Programmiersprache
- plattformunabhängig
- Klasse – Objekt
 - Attribute – Instanzvariablen
 - Verhaltensweisen – Methoden
- Java erlaubt Multithreading
- Zahlreiche Klassenbibliotheken
- Entwicklungsumgebung
 - Java SDK (incl. JRE)
 - Eclipse

■ Processor 14 - Calls to Java

- ▶ Name Association Syntax:

```
(class signature) 14 NA 'surrogate member'
```

```
class ←! 'javax/sound/midi/MidiSystemé
signature ←! '()Ljavax/sound/midi/Synthesizer;'
member ←! 'getSynthesizer'
```

(getSynthesizer ist danach eine Funktion)

■ P14 Hilfsfunktionen

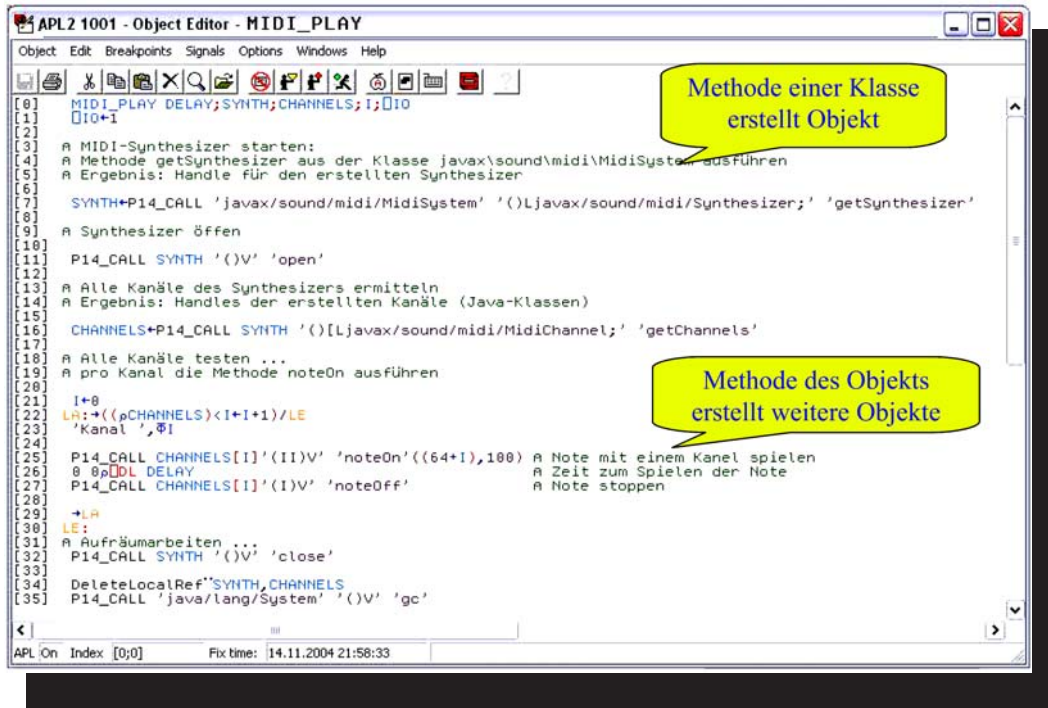
... aus 2 DEMOJAVA

- P14_ASSOC
P14_ASSOC class signature member
statt „14 NA“
- P14_CALL
P14_CALL class signature member parameter
führt Java-Methode aus
- P14_MAKE
P14_MAKE class signature parameter
erstellt eine Instanz einer Klasse

■ APL2 macht Musik

- ▶ „Calls to Java“ – Beispiel: MIDI_PLAY

APL - Journal 2004, 23. Jg., Heft 2



Tor zur Java Welt

hilfreiche Java-Klassen und -Applikationen

- Graphik, z.B. VTK
- WebServices (SOAP, XMP-RPC)
- XML (SAX und DOM-Parser)
- ... u.v.m.

Links

- java.sun.com
- www.apache.org
- www.ibm.com

Calling APL2 from Java

Klassen in com.ibm.apl2 (apl2.jar)

- **Apl2interp** - führt APL2-Anweisungen aus
- **Apl2object** - konvertiert Daten zwischen Java und APL2
- **Apl2cdr** - konvertiert Daten zwischen Workspace- und CDR-Formaten (zum Übertragen von Variablen zwischen APL2-Interpretern)
- **Apl2exception** - behandelt Fehler während der Ausführung von APL2-Anweisungen

Ein ganz einfaches Beispiel (siehe APL2 User's Guide, S. 501)

```
Apl2interp Slave = new Apl2interp(new String[] {,-ws",";1m"});
Apl2object Time = Slave.Execute(Apl2interp.QTS);
int[] TimeArray = Time.intarrayValue();
Time.Free();
Slave.Stop();
System.out.println(„The time is „ + TimeArray[4] + ‘:’ +
TimeArray[5] + ‘.’);
```

... was zu beachten ist

Definierte APL2-Funktionen ausführen:

- ➔ nur mit Funktionen eines Namespaces

- **Apl2interp** Objekt instantiieren
- Funktion aus Namespace assoziieren
- Funktion (evtl. mit Argumenten) ausführen
- evtl.: Ergebnis weiterverarbeiten
- Konvertierung von Argumenten und Ergebnis mit **Apl2object**

Wozu das alles?

- APL2 Anwendungen können in Java-Projekten genutzt werden
- Stärken von APL2 stehen Java Entwicklern zur Verfügung
- Java als „Connector“ zwischen APL2- und anderen Anwendungen
- Nutzung von APL2 innerhalb Java-Anwendungen (z.B. SAX XML)
- APL2 unter Java Server Pages
- „Multithreaded APL2“

APL2Env (Rückblende)

... zeigt aktuelle APL2-Umgebung an

- Bibliotheksdefinitionen
- Weitere APL2-Umgebungsvariablen
- APL2 Aufrufsoptionen
- Prozessor 11 Namen



Aber: Session Manager ist während der Anzeige gesperrt

Die Dienstleister

➔ Lösung:

Multithreaded APL2

Lösung: Ausführung in eigenem Thread

PUSH

Name	Größe	Typ	Geändert am
BFF.apl	798 KB	Workspace	15.10.2004 11:17
TOOLS.apl	218 KB	Workspace	04.10.2004 16:12
GRID2.apl	201 KB	Workspace	23.09.2004 09:40
GRID1.apl	447 KB	Workspace	20.09.2004 13:14
GRID.apl	522 KB	Workspace	19.09.2004 16:34
DDEEXCEL.apl	673 KB	Workspace	06.09.2004 16:59
TOOLS.ans	173 KB	Namespace	03.09.2004 20:06
TEMP1.apl	188 KB	Workspace	03.09.2004 17:21
TEMP.atf	5 KB	Transfer File	03.09.2004 17:14
TEMP.apl	188 KB	Workspace	03.09.2004 16:43
CNSUTIL2.apl	225 KB	Workspace	13.08.2004 09:33
CNSUTIL.apl	1.804 KB	Workspace	13.08.2004 09:25
SYMBOLS.apl	1.936 KB	Workspace	28.06.2004 15:32
UTIL01.apl	368 KB	Workspace	18.06.2004 12:53
FTPHOST.apl	11.219 KB	Workspace	16.06.2004 21:26
TOOLS2.apl	289 KB	Workspace	06.06.2004 12:22
WSDOCWIN.apl	1.892 KB	Workspace	12.05.2004 12:10
WSDOCWIN.atf	178 KB	Transfer File	30.04.2004 17:14
DEMO.ans	7 KB	Namespace	29.04.2004 13:13
UTIL01.ans	368 KB	Namespace	28.04.2004 14:50
DSL1ST.ans	101 KB	Namespace	28.04.2004 12:00
GETAUTH.ans	51 KB	Namespace	28.04.2004 12:00
WSTRANS.ans	401 KB	Namespace	28.04.2004 12:00
TOOLS1.ans	147 KB	Namespace	27.04.2004 13:35
TOOLS1.apl	147 KB	Workspace	27.04.2004 13:35
DATETIME.apl	130 KB	Workspace	27.02.2004 12:10
COMMUTE.apl	8 KB	Workspace	26.02.2004 15:29
DDEUTIL.apl	37 KB	Workspace	14.01.2004 16:34
TSVGL.atf	0 KB	Transfer File	14.01.2004 15:05

```

APL2 1001 - Object Editor - PUSH
Object Edit Breakpoints Signals Options Windows Help
[0] Apl2exec(FLoc PUSH)x;StartedThread;Namespace;Function;RArg;L
[1] Ap Ausführung einer APL2-Funktion in einem neuen Thread
[2] Ao DPC/AH, 17.06.2004
[3] Ar I0+A (C1 PUSH) A
[4] Ad FLoc: Funktionsname
[5] Ad x,y: Argumente für die auszuführende Funktion
[6] Ad Apl2exec: Handle des Apl2Exec-Objektes
[7]
[8] DES(1#3 11 DNa 'ATR')/1 5
[9]
[10] Namespace+'E:\APL-WS\UTILES\TOOLS.ans'
[11] Function+FLoc
[12]
[13] →(2#DNC 'y')/DVA
[14]
[15] RArg+P14_MAKE 'com/ibm/apl2/Apl2cdr' '([B]V'(CATR x)
[16] Apl2exec+P14_MAKE 'Apl2Exec' '(Ljava/lang/String;Ljava/lang/St
[17] →8
[18]
[19] DVA:
[20] RArg+P14_MAKE 'com/ibm/apl2/Apl2cdr' '([B]V'(CATR x)
[21] LArg+P14_MAKE 'com/ibm/apl2/Apl2cdr' '([B]V'(CATR y)
[22] Apl2exec+P14_MAKE 'Apl2Exec' '(Ljava/lang/String;Ljava/lang/St
[23]
    
```

Apl2Exec

```

Apl2Exec.java - Editor
Datei Bearbeiten Format Ansicht 2
public void run()
{
    try
    {
        Apl2Interp slave = new Apl2Interp(new String[] {"-ws","10m,200m,10m"});
        slave.Associate(Apl2Namespace,11,Apl2Function);
        // Attribute der auszuführenden Funktion (funktioniert nicht)
        Apl2Object Function = new Apl2Object(slave,Apl2Function);
        Apl2Object EIns = new Apl2Object(slave,1);
        Apl2Object Attributes = slave.Execute(EIns,Apl2Interp.QAT,Function);
        int[] AttrArray = Attributes.intarrayvalue();
        // Funktion im Namespace ausführen
        if (Apl2RightArg==null) // n1ladisch
        {
            System.out.println("n1ladisch:"+Apl2Function);
            Apl2Object Result=slave.Execute(Apl2Function);
            System.out.println("done:"+Apl2Function);
            if (Result!=null){
                Apl2Result =Result.cdrvalue();
                Result.Free();
            }
        }
        else
        {
            if (Apl2LeftArg==null) // monadisch
            {
                System.out.println("monadisch:"+Apl2Function);
                Apl2Object Right = new Apl2Object(slave,Apl2RightArg);
                Apl2Object Result=slave.Execute(Apl2Function,Right);
                Right.Free();
                System.out.println("done:"+Apl2Function);
                if (Result!=null){
                    Apl2Result =Result.cdrvalue();
                    Result.Free();
                }
            }
        }
    }
}
    
```

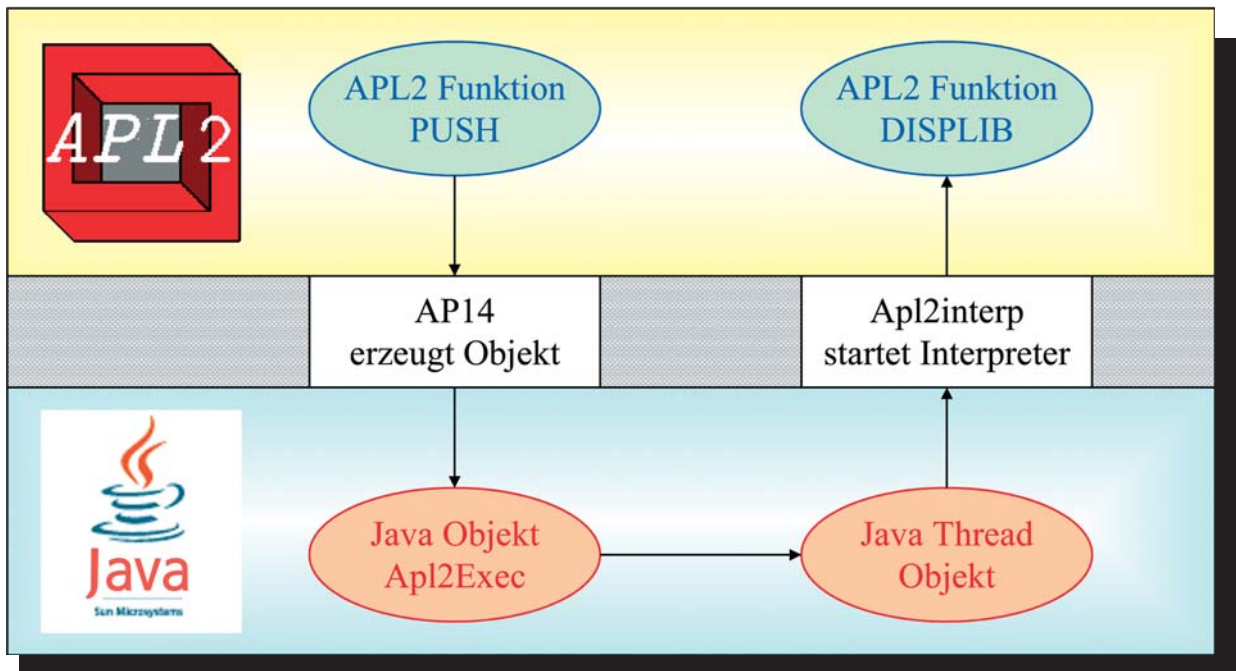
aber: kein Multithreading unter APL2

Java unterstützt Multithreading

➔ APL2 – Java – APL2:

Aus 1 mach 2 ...

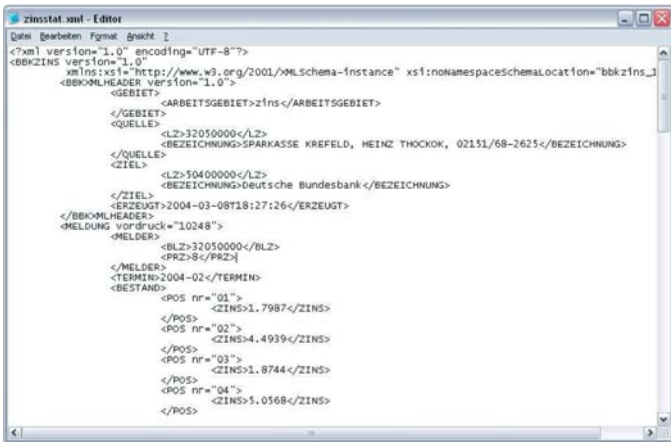
Vorwärts und rückwärts



... und mehr

- weitere Tools: DISPLIB, LastFns, DISPVAR
- Parallele Ausführung von Funktionen
- APL2-Server

... und XML-Parser unter APL2



Tor zur Java Welt

- Bemerkenswertes:
 - Session ID: unter Java gestartete Interpreter erhalten normale Session ID
 - Classpath
- Java erlaubt Multithreading: „multithreaded APL2“, parallele Ausführung

```
( < 'COUNT' ) PUSH * 10 20 15 5
```

Tor zur Java Welt

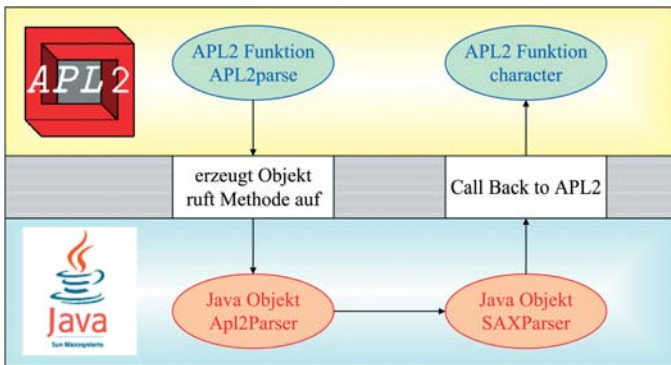


„Multithreading“ für APL2 (nah und fern)



Java als „Connector“ zu anderen Anwendungen

APL2 und SAX



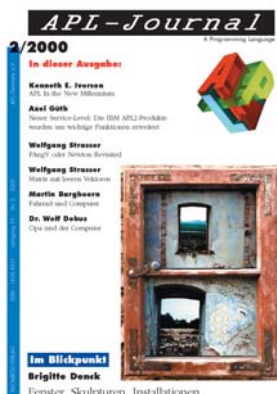
- Anschluß an aktuelle Trends: XML, Webservice u.a.

Kontakt:

Axel Holzmüller,
eMail: axel.holzmueller@dpc.de

Dittrich & Partner Consulting GmbH, Kieler Str.17, D-42697 Solingen, Tel. 0212-26066-0, Fax: -26066-66, eMail: info@dpc.de

APL - Germany e.V.



<http://www.apl-germany.de>



Romilly Cocking

Arrays to the Rescue!

SmartRec-A SmartArrays application at BNP Paribas

SmartRec

- A Configurable Reconciliation Engine
- Implemented in Java
- Based on SmartArrays

SmartArrays

- Very fast array engine
- Inspired by APL
- Developed by James Wheeler
 - APL*Plus
- And Jim Brown
 - APL2
- For C++, Java, .Net developers

What is Reconciliation?

- A Business Process
- Matches and Compares
- Two or more sets of records
 - that should tell the same story

example

- Personal banking
- Comparing cheques with a bank statement

A Cheque



Account	Ch. Number	Date	Amount	Payee
987654321	100208	28/03/06	£10.00	A.N.Uther

A Bank Statement

Date	Payment type and details	Paid out	Paid in	Balance
29 Mar 06	BALANCE BROUGHT FORWARD			2000.00
31 Mar 06	CHQ 100205	13.50		1987.50
01 Apr 06	DD Farnet Insurance	30.00		1957.50
03 Mar 06	CHQ 100207	10.00		1947.50
05 Mar 06	CHQ 100206	100.00		1847.50
29 Mar 06	BALANCE CARRIED FORWARD			1847.50

What banks want to reconcile

- Trade data
 - Internally (Front and Back office)

- Externally (Buyer and seller)
- User data
- Accounting data

Key concerns

- Flexibility
- Reliability and accuracy
- Performance
 - Most banks use RDBMS
 - Volumes of data very large
 - Example - 2.000.000 records vs 2.000.000 records
 - Currently takes 4 hours on RS6000

Where data comes from

- Flat files
- Spreadsheets
- Databases
- XML

First results

- Complex reconciliation
- User data
- 14937 vs. 37314 records
- Old system:

- ❑ ~ 90 minutes on RS6000
- SmartRec: ~90 seconds on this laptop
- ❑ Now ~ 30 seconds using SmartArrays v3 + toolkit

SmartRecstatus

- Will replace existing solution (RAID) throughout BNPP
- Longer term, will replace 3rd party products
- Plan to make SmartReca product (this year)

the technology

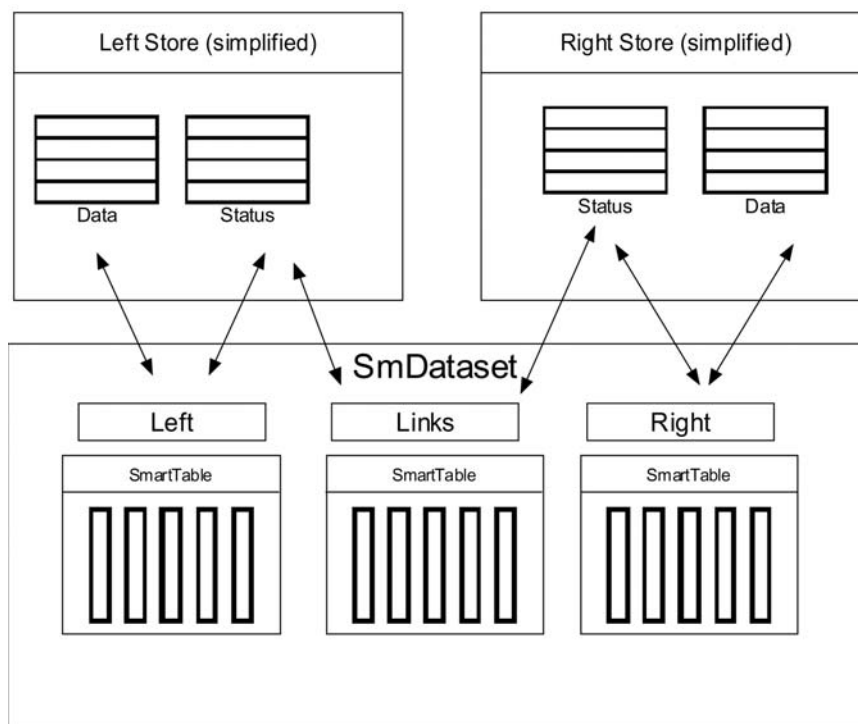
Requirements

- Linking and comparing
- Transformations
- Filtering

Architecture

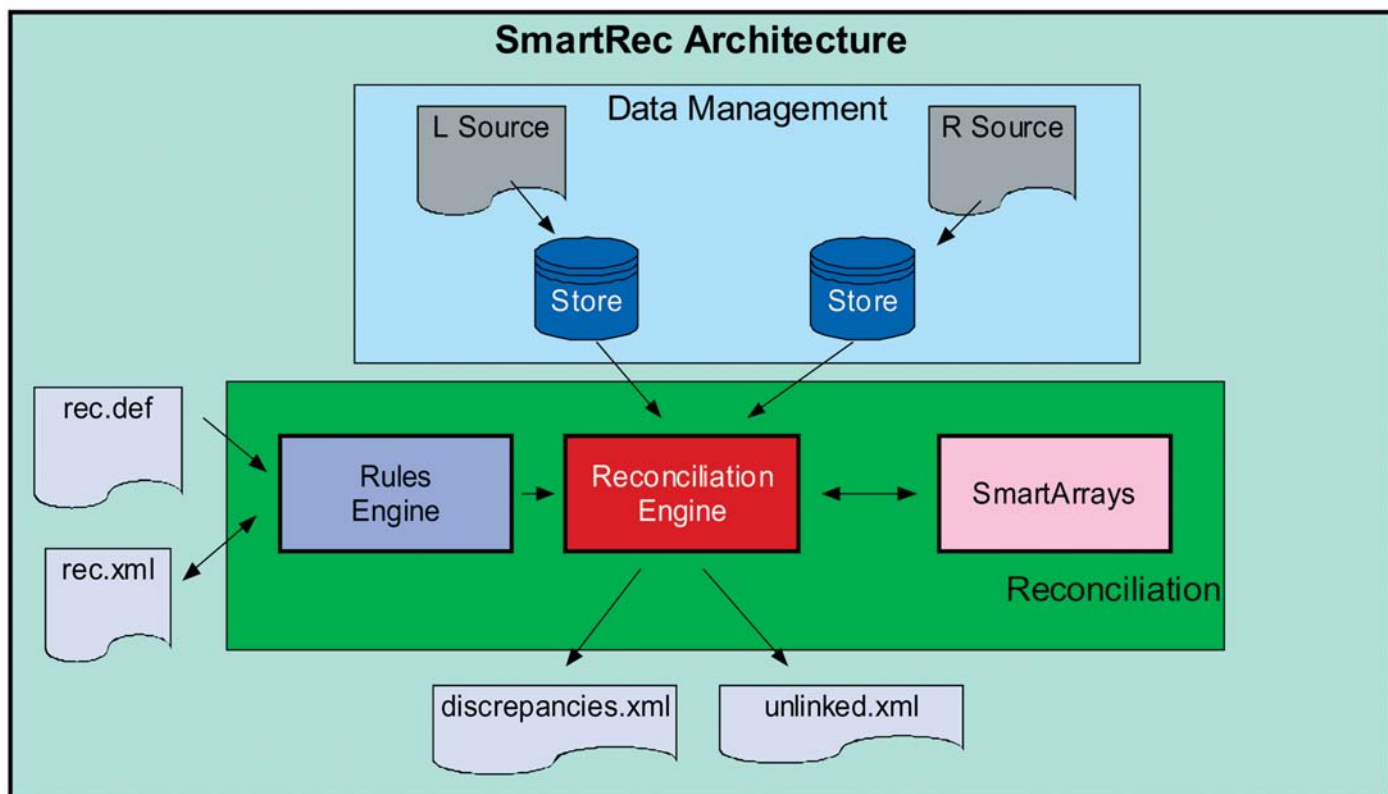
- Compiler
 - ❑ Turns specification into objects that define what is to be done
- Data Manager
 - ❑ Loads data
 - ❑ Remembers unlinked records from previous runs
- Reconciliation Engine
 - ❑ Interprets and executes the reconciliation
 - Filters and Transforms data
 - Links and compares data
 - Creates outputs

From database to array and back



SmartArrays code

```
SmArraycp = keys.eq(SmArray.vector(1).
    rotateAxis(keys, 0)).reduceAxis(Sm.and, 1);
SmArraybv= SmArray.scalar(0).catenate(cp)
    .dropBy(-1);
bv= bv.lt(bv.rotateBy(1));
```



Internals

- 1-1 reconciliation
- 1-m reconciliation
- m-nreconciliation
- Grouping and Joining
- XML reconciliation

Reconciliation specification

- Declarative language
 - (John Scholes would say Izzy)
- Originally created to help me write tests
- When users saw it, they wanted it

Rec.def

```
reconciliation {
  stores {
    leftdocs {
      source delimited separator = ":".
      fields {
        id(),
        sale_price(name="sale price", number),
        cost(name="cost", number),
        cp(name="counterparty")
      }
      working {
        profit(name="we made", number)
      }
      transformations {
        profit = sales - cost.
      }
    }
    rightdocs {
      source delimited separator = ":".
      fields {
        id(),
        profits(name="profit", number),
        cp(name="counterparty")
      }
    }
  }
  model model1 {
    link id to id.
    compare cp to cp.
    compare profit to profits.
  }
}
```

1-1 Reconciliation

Model 1		
Object A1		Object B1
Portfolio = 105	Matching Criterion	Portfolio = 105
Amount = 10,000	Matching Criterion	Amount = 10,000
Currency = GBP	Discrepancy Criterion	Currency = GBP
Location = London	Discrepancy Criterion	Location = Paris

Evolution

- Server based version
- Production version
- Product family
- GUIs
 - for reconciliation definition
 - for results

1-m reconciliation (complex)

Model 4		
Object A5		Object B5
Object A5 ID = YOY	Matching Criterion	Object B5 ID = ZOZ
Portfolio = 109	Discrepancy Criterion	Reference A5 ID = YOY
Amount = 10,000	Discrepancy Criterion	Portfolio = 109
Amount = 10,000	Discrepancy Criterion	Amount = 12,000
Currency = GBP	Discrepancy Criterion	Currency = GBP
Location = London	Discrepancy Criterion	Location = London
Object A5		Object B6
		Object B6 ID = BOB
		Reference B5 ID = ZOZ
Object A5 ID = YOY		no reference to A5 ID
Portfolio = 109	Discrepancy Criterion	Portfolio = 110
Amount = 10,000	Discrepancy Criterion	Amount = 10,000
Currency = GBP	Discrepancy Criterion	Currency = GBP
Location = London	Discrepancy Criterion	Location = London

Conclusions

- BNP Paribas delighted
 - Outstanding Performance
 - Astounding Progress
 - Sees SmartArrays as key to success
- Arrays worked wonders!
 - Now spreading within BNP Paribas
 - Interest from other UK banks

Kontakt:

Romilly Cocking,
<http://www.cocking.co.uk>



TwoView: Neues Display-System für Arbeiten im virtuellen Raum

Neue Technik ermöglicht räumliche Präsenz virtueller Objekte und kann in der Industrie eingesetzt werden

Sankt Augustin/Düsseldorf. Das Fraunhofer-Institut für Medienkommunikation IMK in Sankt Augustin und die Firma Barco haben ein neuartiges, projektorbasiertes Display-System entwickelt, das eine exakte Zusammenarbeit und gemeinsame Interaktion von zwei Benutzern in einer virtuellen Umgebung ermöglicht. Mit dem Display-System, das TwoView, werden die gemeinsame Analyse von komplexen Daten und das kollaborative Engineering im virtuellen Raum entscheidend besser als mit den bisher üblichen Systemen unterstützt, berichtet das IMK.

Im Gegensatz zu konventionellen Display-Systemen wie der CAVE, die nur ein einziges Bild für mehrere Betrachter darstellen, wird beim TwoView-Display für jeden der beiden Benutzer ein eigenes stereoskopisches 3D-Bild erzeugt. Dies ermöglicht es, virtuelle Objekte für jeden Betrachter räumlich absolut korrekt und deckungsgleich darzustellen. "Wenn zwei Benutzer eine Darstellung exakt am selben Punkt im Raum wahrnehmen, wird eine völlig neue Präsenz virtueller Objekte erzielt", erläutert Andreas Simon, der das System am Fraunhofer IMK als Display-Spezialist konzipiert hat. Die "Echtheit" eines virtuellen Objekts lässt sich ganz anders akzeptieren, wenn man beobachten kann, dass eine andere Person diesen Gegenstand genau so sieht wie man selbst. Erst so wird eine wirkliche Zusammenarbeit im virtuellen Raum ermöglicht.

Vergleichbare Displays wurden bisher nur als Forschungsprototypen aufgebaut. Mit dem TwoView-Display verfügt das



L-Shape: Zusammenarbeit im virtuellen Raum (Foto: Fraunhofer IMK)

Fraunhofer IMK erstmals über ein System, das den produktiven Einsatz dieser Technik im industriellen Umfeld ermöglicht. Das TwoView wird zum Beispiel zukünftig im VRGeo-Projekt (<http://www.vrgeo.org>) genutzt, in dem Anwendern aus der Öl- und Gasindustrie die Zusammenarbeit beim Entwurf aufwendiger Installationen, etwa bei der Planung von Bohrlöchern, ermöglicht wird und in dem die Zusammenarbeit von Experten bei der Analyse komplexer geowissenschaftlicher Datensätze mit Hilfe von Virtual-Reality-Technologie verbessert wird. Weitere unmittelbare Anwendungsfelder des TwoView-Displays sind das kollaborative Engineering in der Automobi-

lindustrie und auch die Diagnose in der Medizintechnik.

Kontakt: Fraunhofer-Institut für Medienkommunikation IMK, Schloss Birlinghoven, D-53754 Sankt Augustin, Thorsten Holtkämper, eMail: thorsten.holtkaemper@imk.fraunhofer.de, Tel: 02241/14-3419, Dr. Manfred Bogen, eMail: manfred.bogen@imk.fraunhofer.de, Tel: 02241/14-2367, Internet: www.imk.fraunhofer.de/de/twoview

Barco GmbH, Tiefenbroicher Weg 35/A2, D-40472 Düsseldorf, Internet: www.barco.com.

Quantencomputer aus gewöhnlichen Transistoren

Kleiner Elektronen-Spin soll Rechentechnik revolutionieren

Los Angeles. Forschern der Universität von Kalifornien (UCLA) ist es gelungen, den Spin eines Elektrons in einem gewöhnlichen Transistor-Chip zu verändern. Dabei entdeckten sie, dass damit auch die Ladung wechselte. Ihre Versuche gelten als wichtiger Schritt für die Konstruktion von Quantencomputern.

Um den Elektronen-Spin zu ändern nutzten die Wissenschaftler eine Radio-Frequenz im Mikrowellenbereich. „Unser Versuch hat gezeigt, dass ein ganz einfacher Transistor, wie er in einem gewöhnlichen PC oder Mobiltelefon verwendet wird, für

den Bau von Quantencomputern geeignet ist“, erklärte UCLA-Professor Hong Wen Jiang. Durch das gelungene Experiment soll der Bau von Quantencomputern weit realistischer werden, da der Spin eines einfachen Elektrons schließlich ein Q-Bit, den Grundbaustein eines Quantencomputers repräsentiert.

Bevor sich der Erfolg einstellte, hatten die Forscher um UCLA-Professor Hong Wen Jiang und den Doktoranden Ming Xiang mit Mio. von Elektronen experimentiert. Für den gelungenen Versuch arbeiteten Jiang und Xiang mit gewöhnli-

chen Transistoren bei minus 400 Grad Fahrenheit. In Zukunft soll die Technik aber auch bei normalen Temperaturen funktionieren. Dadurch soll die kommerzielle Nutzung vereinfacht werden. (pte)

Ausführliche Informationen: <http://www.newsroom.ucla.edu/page.asp?RelNum=5356>

Kontakt: UCLA, 405 Hilgard Avenue, Box 951361, Los Angeles, CA 90095-1361, Mr. Stuart Wolpert, eMail: stuartw@college.ucla.edu, Internet: <http://www.ucla.edu>.

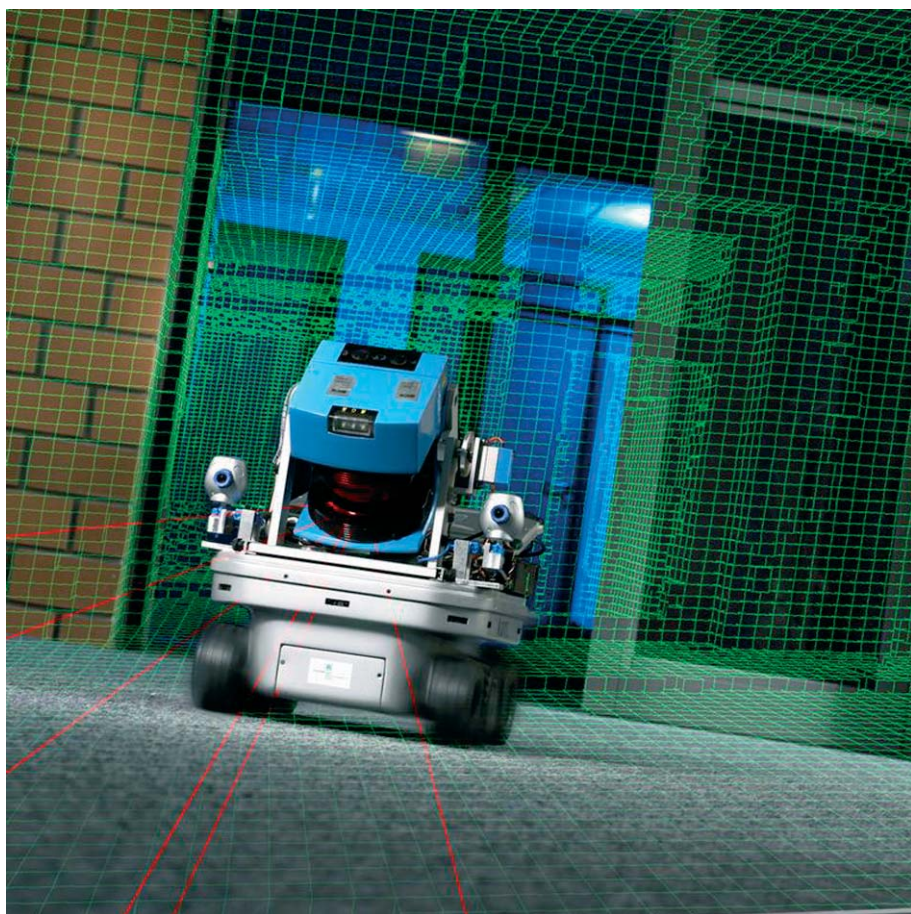
Roboter mit Köpfchen

Intelligente Systeme von morgen sollen wie Menschen lernen und Probleme lösen können

Sankt Augustin. Roboter haben ein schlechtes Image: Sie gelten als eher beschränkt und unkreativ. Doch künftig sollen sie intelligent und flexibel werden. Im EU-Projekt MACS bringt ein internationales Forscherteam ihnen funktionsorientiertes Denken bei. Die Roboter von morgen sollen wie Menschen lernen und Probleme lösen können.

Wann ist eine Maschine intelligent? Wenn sie genau das tut, wofür sie gebaut wurde - beispielsweise Putzen oder Rasen mähen? Oder beginnt Intelligenz erst jenseits der vorprogrammierten Handlungstätigkeiten? „Ein wichtiges Kriterium ist Flexibilität“, sagt Dr. Erich Rome, Abteilungsleiter am Fraunhofer-Institut für Autonome Intelligente Systeme AIS und Leiter des EU-Projekts „Multi-sensory Autonomous Cognitive Systems Interacting with Dynamic Environments for Perceiving and Using Affordances“, kurz MACS. „Für einen Menschen ist es selbstverständlich, flexibel auf äußere Umstände zu reagieren. Sie denken immer funktional, also am Ergebnis orientiert“, erklärt Rome. „Roboter hingegen arbeiten bisher objektorientiert.“ Wenn eine Maschine darauf programmiert ist, einen Stuhl zu bringen, dann sucht sie ausschließlich nach einer Sitzgelegenheit mit vier Füßen und einer Lehne. Auf die Idee, dass auch ein leerer Getränkekasten den Anforderungen genügen könnte, kommt sie nicht.

„Um Maschinen funktionales Denken beizubringen, muss man ihnen die Gelegenheit geben, die Dinge zu begreifen wie kleine Kinder“, ergänzt der Informatiker Rome. „Sie müssen ausprobieren und lernen können, was man mit einem Gegenstand alles machen kann.“ Der amerikanische Wahrnehmungspsychologe James Gibson hat dafür den Begriff „Affordanz“ eingeführt: Er ist abgeleitet vom englischen Wort „afford“ - zu deutsch „anbieten“ oder „gewähren“. Eine Affordanz ist eine Handlungsmöglichkeit, die die Umwelt einem Menschen bietet, und die er erkennen und nutzen kann. „Im EU-Projekt MACS bauen wir auf Gibsons Theorie auf, um neue Ansätze für intelligente und lernfähige Roboter zu entwickeln“, so Rome. Der Fraunhofer-Forscher und sein Team arbeiten dabei eng mit Wissenschaftlern der schwedischen Linköpings Universität, der Forschungsgesellschaft Joanneum Research und der Studiengesellschaft für Kybernetik in Österreich sowie der türkischen Middle East Techni-



Im EU-Projekt MACS soll der Roboter KURT 3D spielerisch seine Umgebung kennen lernen. (Foto: Dieter Klein)

cal University zusammen. Die EU unterstützt das Projekt mit 1,9 Millionen Euro. Drei Jahre haben die Forscher Zeit, um KURT 3D, einem flexibel einsetzbaren Roboter vom Fraunhofer-Institut in St. Augustin, das funktionale Denken beizubringen. Derzeit wird ein Übungsraum konzipiert - eine Art Spielzimmer, in dem KURT 3D mit seinem magnetischen Greifarm nach Herzenslust experimentieren kann. Er wird voraussichtlich ein ganzes Sortiment von Metallbausteinen bekommen, die sich anheben, stapeln oder verschieben lassen. Beispielsweise große und kleine Dosen, Rampen und Schwellen. Wenn der Roboter mit diesen Objekten „spielt“, lernt seine Steuerungssoftware ständig dazu. Nach einiger Zeit soll KURT 3D seinen Aktionsradius selbstständig vergrößern. Dazu muss er allerdings erst einmal einiges über seine Umwelt herausfinden: beispielsweise dass sich eine Tür nur öffnet, wenn man mehrere schwere Gegenstände auf einen Schalter stellt; oder dass sich Schwellen mit Hilfe von Rampen über-

winden lassen. „Im EU-Projekt werden wir noch an den Grundlagen arbeiten“, erklärt Rome. „Unser Ziel ist es dabei, zu zeigen, dass funktionale Roboter wie Menschen lernen und Probleme lösen können.“ Langfristig werden die neuen, flexiblen Denkstrukturen nach Ansicht des Projektleiters die Roboterwelt verändern: „Die Grundlagenforschung von heute wird in die Entwicklung der nächsten und übernächsten Roboter-Generation mit einfließen. Bisher sind Objekt- und Funktionsorientierung separate Ansätze. Tatsächlich könnten sich die beiden Ansätze jedoch ergänzen. Je nach Situation kann es für einen Roboter sinnvoll sein, nach Handlungsalternativen zu suchen oder stur nach Plan vorzugehen.“

Kontakt: Dr. Erich Rome, Tel. 02241-1426 83, eMail: erich.rome@ais.fraunhofer.de. Fraunhofer-Institut für Autonome Intelligente Systeme (AIS), Schloss Birlinghoven, D-53754 Sankt Augustin, Tel. 02241-14-27 56, -26 78, Fax: -14-23 84, Internet: www.ais.fraunhofer.de.

Allgemeine Informationen

(Stand Herbst 2004)



APL-Germany e.V. ist ein gemeinnütziger Verein mit Sitz in Düsseldorf. Sein Zweck ist es, die Programmiersprache APL, sowie die Verbreitung des Verständnisses der Mensch-Maschine Kommunikation zu fördern. Für Interessenten, die zum Gedan-

ken austausch den Kontakt zu anderen APL-Benutzern suchen, sowie für solche, die sich aktiv an der Weiterverbreitung der Sprache APL beteiligen wollen, bietet APL-Germany den adäquaten organisatorischen Rahmen.

Auf Antrag, über den der Vorstand entscheidet, kann jede natürliche oder juristische Person Mitglied werden. Organe des Vereins sind die mindestens einmal jährlich stattfindende Mitgliederversammlung sowie der jeweils auf zwei Jahre gewählte Vorstand.

Vorstand

Vorsitzender

Dieter Lattermann
Rheinstraße 23
69190 Walldorf
Tel. (06227) 63469
eMail: Dieter_Lattermann@t-online.de

Schriftführer

Conrad Hoesle-Kienzlen
CHK IT-Consulting
Augustenstr. 57
70178 Stuttgart
Tel. 0172 722 04 48
Fax 0711 674 38 53
eMail: conrad@hoesle.net

2. Vorsitzender

Martin Barghoorn
(Technische Universität Berlin)
Lückhoffstr. 8
14129 Berlin
Tel. (030) 314 24392
Fax: (030) 314 25901
eMail: martin@barghoorn.de
<http://stat.cs.tu-berlin.de/~barg/>

Schatzmeister

Jürgen Beckmann
Im Freudenheimer Grün 10
68259 Mannheim
eMail: jbecki@onlinehome.de

Beitragssätze

Ordentliche Mitglieder:

Natürliche Personen 31,00 Euro p.a.
Studenten / Schüler 10,00 Euro p.a.

Außerordentliche Mitglieder:

Juristische /
natürliche Personen 511,00 Euro p.a.

Bankverbindung:

BVB Volksbank eG Bad Vilbel
BLZ 518 613 25, Konto-Nr. 523 2694

Hinweis:

Wir bitten alle Mitglieder, uns *Adressenänderungen* und *neue Bankverbindungen* immer sofort mitzuteilen.

Geben Sie bei *Überweisungen* den *Namen* und/oder die *Mitgliedsnummer* an.

Einzugsermächtigung

Mitglieds-Nr.: _____

Ich erkläre mich hiermit widerruflich damit einverstanden, daß APL Germany e.V.

den jeweils gültigen Jahres-Mitgliedsbeitrag von meinem unten angegebenen Konto abbucht.

Einen eventuell bestehenden Dauerauftrag habe ich bei meiner Bank gelöscht.

Bankbezeichnung: _____

BLZ: _____

Konto-Nr.: _____

Datum: _____

Unterschrift: _____