A. Sorgatz: Dynamic Modules. User's Manual and Programming Guide for MuPAD 1.4, Springer-Verlag, Berlin Heidelberg 1999, ISBN 3-540-656043-1

Verglichen mit den Anfängen der Computeralgebra werden heute die Systeme im häufiger in Verbindung mit anderen Programmen oder Programmbibliotheken eingesetzt. Hier spiegelt sich der Trend wieder, daß Computeralgebrasysteme immer mehr als nur ein Teil in einer komplexen *problem solving environment* angesehen werden, zu der z.B. auch Numerikbibliotheken oder Visualisierungstools gehören. Angesichts dessen ist es nicht verwunderlich, daß die meisten Systeme sich öffnen und Möglichkeiten zur Kommunikation mit anderen Programmen anbieten; als Beispiele seien nur MathLink und MathEdge für Mathematica bzw. Maple erwähnt.

Der klassische Weg zur Realisierung einer solchen Schnittstelle geht über *interprocess communication* (IPC). Dieser Ansatz führt aber zu einem beträchtlichen Kommunikationsoverhead, da Daten zwischen den verschiedenen Programmen explizit ausgetauscht werden müssen. Dies macht IPC vor allem dann langsam, wenn entweder sehr große Mengen an Daten transferiert werden (ganz abgesehen davon, daß diese Daten dann doppelt im Speicher gehalten werden müssen) oder allgemeiner die Verarbeitungszeit verhältnismäßig klein gegenüber der Kommunikationszeit ist.

Andreas Sorgatz von der Paderborner *MuPAD*-Gruppe hat in *MuPAD* einen anderen Ansatz zur Integration von Softwarepaketen realisiert: dynamische Module. Dynamisch bedeutet hierbei, daß diese Module zur Laufzeit zu *MuPAD* gelinkt werden können. Sie werden damit zu einem Teil des *MuPAD*-Prozesses und teilen sich vor allem den Speicher mit *MuPAD*. Damit muß kein Datentransfer mehr erfolgen. Um die Möglichkeiten dieses Ansatzes auszuschöpfen, benötigt der Entwickler dynamischer Module allerdings einige Kenntnisse von Interna von *MuPAD*.

Rein technische Fragen der Realisierung dynamischer Module wurden bereits in einem früheren Buch des Autors behandelt (A. Sorgatz, Dynamische Module, *MuPAD*-Reports, Teubner 1996, besprochen im Zentralblatt Zbl. 970.06779). Was bisher noch fehlte, war eine Beschreibung der dynamischen Module aus der Sicht des Anwenders, der entweder mit vorgegebenen Modulen arbeiten oder aber eigene Module entwickeln will. Diese Lücke wird durch das vorliegende Buch geschlossen. Es basiert auf der aktuellen *MuPAD*-Version 1.4.1, die auch auf der beiliegenden CD enthalten ist.

Wie der Untertitel des Buches bereits klar ausdrückt, handelt es sich hier in erster Linie um ein Handbuch, d.h. die wenigsten Leser werden es von vorne bis hinten durchlesen sondern es als Nachschlagewerk benutzen. Nach einigen einführenden Kapiteln, in denen die zugrundeliegenden Konzepte vorgestellt werden, wird das Buch denn auch sehr technisch. Dies gilt insbesondere für das zentrale Kapitel 6, das das *MuPAD* Application Programming Interface (MAPI) beschreibt. Das MAPI stellt eine umfangreiche Sammlung von vordefinierten Typen und Prozeduren dar, die es dem Entwickler von dynamischen Modulen gestatten, *MuPAD*-Objekte zu erzeugen bzw. zu manipulieren. Weitere Kapitel beschreiben u.a. die interne Struktur von *MuPAD*, den Modulgenerator zum Kompilieren und Linken von dynamischen Modulen oder geben Details zu speziellen Betriebssystemen, Programmiersprachen, shared/static libraries etc.

Erfreulicherweise enthält das Buch sehr viele Beispiele (oft mit komplettem Programmcode), die dem Anfänger den Einstieg in die Welt der dynamischen Module sehr erleichtern. Diese reichen von der einfachen Interaktion mit dem Betriebssystem über die Einbindung numerischer Bibliotheken bis hin zur Fensterprogrammierung unter Motif. Auf der CD finden sich auch noch einige größere Beispiele wie die Anbindung von Singular oder GB/RealSolving an *MuPAD*. Auch für eine Reihe von Kommunikationsprotokollen wie MP oder PVM existieren bereits dynamische Module.

Wer in *MuPAD* mit dynamischen Modulen arbeiten will, wird um dieses Buch nicht herumkommen. Die bereits realisierten Anwendungen zeigen das große Potential dieses Ansatzes zur Verknüpfung verschiedenster Programmsysteme. *MuPAD* bietet hier weit flexiblere Möglichkeiten als andere Computeralgebrasysteme; es liegt nun an den Benutzern, diese auch auszunutzen!

Werner M. Seiler (Mannheim)