

Engineering Desktop Anwendungen in der Tragwerksplanung auf der Grundlage der COM-Technologie

Horst Werkle und Hartmut Pleßke

Institut für angewandte Forschung, Fachhochschule Konstanz

1 Einführung

Computer heißt Rechner und prozedural durchzuführende Rechenaufgaben waren lange Zeit die originäre Aufgabe dieses Gerätes. Im Bereich der Baustatik, aus dem Konrad Zuse vor etwa 80 Jahren seine Motivation zur Entwicklung des ersten Computers bezog, ist heute eine Fülle von Programmen für die typischen Aufgaben der Tragwerksplanung mit einem fest vorgegebenen Rechenablauf verfügbar. Wie aber sieht es aus, wenn nicht standardisierbare Aufgaben zu bearbeiten sind? Dies können etwa neuartige Nachweise oder Voruntersuchungen sein, die neben der Rechenfunktionalität das Erstellen von Text und Skizzen erfordern. Hierfür bietet sich Office-Software und für statische Nachweise insbesondere Tabellenkalkulationsprogramme und Computeralgebra-Systeme an. In der Praxis ist das Arbeiten mit diesen allgemeinen Tools allerdings häufig ineffizient, wenn spezielle Module für die Tragwerksplanung fehlen.

Im Rahmen des F&E-Projektes EDATRA (*Engineering Desktop Anwendungen in der Tragwerksplanung*) wurden derartige Module auf der Grundlage eines Frameworks für die Tragwerksplanung entwickelt [WER02]. Das Framework umfasst die für die Tragwerksplanung erforderlichen komplexen Funktionalitäten und kann mit Hilfe der Komponententechnologie von unterschiedlichen Anwendungen genutzt werden. Ziel dieser Entwicklung ist eine Engineering Desktop Umgebung auf dem Computer, bei der nichtstandardisierbare Untersuchungen in der Tragwerksplanung ebenso einfach, aber effizienter, sicherer und besser dokumentiert als mit Bleistift und Papier erfolgen können.

2 Framework für die Tragwerksplanung

2.1 Softwarekonzept

Das Softwarekonzept von EDATRA folgt dem modernen Trend in der Softwareentwicklung nach verteilten Systemen. Es besteht einerseits darin, das Framework mit seinen unterschiedlichen statischen Berechnungsmethoden, Nachweisen und Bemessungsaufgaben für die derzeit am stärksten verbreiteten Office-Anwendungen und die gängigen Computeralgebra-systeme nutzbar zu machen und andererseits darin, es darüber hinaus für ein möglichst breites Spektrum von Statiksoftware dienbar zu halten. Die dafür verwendeten Elemente sollten sowohl plattform- wie auch programmiersprachenunabhängig gestaltet sein.

Wie die Erfahrungen in der Softwareentwicklung der letzten Jahre zeigen, reicht es keinesfalls aus, die Software eines derart konzipierten Frameworks in einer objektorientierten Programmiersprache zu implementieren. Dem bis dato unerfüllten Traum der Softwareentwicklung als ein Zusammensetzen geeigneter Software-Bausteine (Komponenten) kommt heute die Komponententechnologie bereits recht nahe. Sie zielt auf eine Trennung von Anwendungslogik und programmtechnischen Details, die unabhängig voneinander entwickelt werden können. Als Hilfsmittel dienen so genannte Komponentenmodelle, die sich auf Unterstützung komponentenbasierter Softwaresysteme konzentrieren und umfassenden Service bieten, welcher die Entwicklung und Ausführung der Komponenten vereinfacht. Unter einer Komponente versteht man ein Stück Software in binärer Form, das eine kohärente

Funktionalität bietet, die mittels einer oder mehrerer standardisierter Schnittstellen (Interfaces) beschrieben wird, um mit anderen Komponenten zusammenzuarbeiten. Das Komponentenmodell formuliert den strukturellen Rahmen mit entsprechender Infrastruktur für die Entwicklung und Ausführung der Komponenten. [GRU00].

Momentan sind für Middleware-Produkte (Komponentenmodelle) vor allem die Marktanbieter CORBA, COM/DCOM, Enterprise Java Beans und das .NET – Framework zu nennen. Für EDATRA bot sich das *Component Object Model (COM)* aus dem Hause Microsoft vor allem deshalb an, weil im Bauwesen derzeit das Betriebssystem MS-Windows in der Tragwerksplanung am weitesten verbreitet ist und die Software von EDATRA ebenfalls unter MS-Windows entwickelt wurde. COM besteht aus einer Reihe von Systemdateien, die auf einer unteren Ebene von MS-Windows interagieren und legt fest, auf welche Weise die binären Softwaremodule zusammenwirken. Es spezifiziert zudem die Regeln, wie die COM-Komponenten erstellt werden, damit sie dem vorgegebenen binären Format entsprechen, fordert für deren Implementierung jedoch keine bestimmte Programmiersprache.

Ein zu COM alternatives Komponentenmodell ist die *Common Request Broker Architecture (CORBA)*, worunter man aber kein reines Produkt, sondern ausschließlich eine Client-/Server Architektur versteht, die sogenannte *Object Management Architecture (OMA)*. Sie stellt durch das Objektmodell und das Referenzmodell die beiden Säulen zur Verfügung, auf denen CORBA aufbaut [HOF01]. Die große Leistungsfähigkeit von CORBA kommt insbesondere bei der Entwicklung und Integration komplexer verteilter objektorientierter Anwendungen in heterogenen Umgebungen zum Tragen.

Obwohl sich die zur Beschreibung der Schnittstellen dienende *Microsoft Interface Definition Language (MIDL)* stark an C⁺⁺ anlehnt, wurde die Komponentenbibliothek des EDATRA-Frameworks in JAVA implementiert. Die Vorteile von JAVA gegenüber C⁺⁺ liegen vor allem in der Plattformunabhängigkeit, der automatischen Feldgrenzen-Überwachung¹, der Modularisierung durch Pakete, dem Typecast² und der Thread-Synchronisation³. Zudem gehören bei JAVA umfangreiche Klassenbibliotheken zur Sprache, die Einbettung in Webseiten ist bequem über *Applets* möglich und die Sicherheit zur Laufzeit ist größer als bei C⁺⁺.

2.2 Funktionalität des Frameworks

Das Framework soll die für typische Aufgaben der Tragwerksplanung erforderlichen Funktionalitäten zur Verfügung stellen. Diese umfassen die Bereiche der Baustatik, des Stahlbetonbaus, des Stahlbaus und Holzbaus, des Mauerwerkbaus, der Geotechnik und der Bauphysik. Dabei sollten die im Framework implementierten Klassen mit einer möglichst großen Allgemeinheit ausgestattet sein. So ist die Klasse für Durchlaufträger beispielsweise für Berechnungen nach Theorie II-ter Ordnung und elastische Bettung ausgelegt. Sie kann damit auch für Stützenberechnungen oder zur Ermittlung der Schnittgrößen elastisch gebetteter Träger angewandt werden.

Einen Überblick über die Funktionalität der implementierten Klassen des Frameworks gibt Tabelle 1. Darüber hinaus ist eine Erweiterung durch andere Funktionalitäten, etwa aus dem Bereich des Mauerwerkbaus, der Bauphysik und der Geotechnik sinnvoll.

¹ Bei jedem Zugriff auf ein Feld wird kontrolliert, ob der Index richtig ist

² Die Konvertierung der Datentypen ist bei Java meist problemlos möglich, da alle Objekte von *Object* abgeleitet sind

³ Unter einem Thread versteht man leicht zu erzeugende Ausführungsstränge, die unabhängig voneinander arbeiten können.

Bereich	Aufgabe
Statik	Durchlaufträger / Stütze / Elastisch gebetteter Balken
	Fachwerk 2D
	Rahmen 2D
	Platten (Rechteckplatte)
Stahlbetonbau	Biegebemessung (ω -Verfahren, k_d -Verfahren)
	Schubbemessung (Bügel, schräge Stäbe)
Stahlbau	Spannungsnachweise / Bemessung / zulässige Schnittgrößen
	Biegeknicken / Biegedrillknicken
	Profiltabellen
Holzbau	Spannungsnachweise / Bemessung / zulässige Schnittgrößen
	Stabilitätsnachweise

Tabelle 1: Funktionalität des Frameworks

2.3 Komponentenbibliothek

Das EDATRA-Framework wurde vollständig objektorientiert in Java entwickelt. Es besteht aus mehreren Klassenbibliotheken für die unterschiedlichen Bereiche der Tragwerksplanung. Aus diesen werden COM-DLL's erstellt und diese dann in die Registry von Windows eingetragen. Der Zugriff auf eine COM-DLL erfolgt über eigens definierte COM-Klassen und deren Methoden.

Die Umsetzung des Frameworks sei am Beispiel des Durchlaufträgers skizzenhaft erläutert. Die COM-Klasse für den Durchlaufträger ist *DLTV2*. Das statische System wird mit *set*-Methoden beschrieben, die Ergebnisse mit *get*-Methoden übergeben. Hierdurch wird die Client-Programmierung sehr flexibel und leicht handhabbar. Die statische Berechnung erfolgt in der *berechnen*-Methode von *DLTV2*. Sie beruht auf dem Übertragungsmatrizenverfahren der Baustatik. Der Zustandsvektor \underline{Z} wird in Abhängigkeit von den Anfangsunbekannten aufgestellt und durch sukzessive Anwendung der Beziehung

$$\underline{Z}_{i+1} = \underline{A}_i \cdot \underline{Z}_i + \underline{L}_i \quad (1)$$

über die Abschnitte und Zwischenpunkte eines Trägerfeldes übertragen. Die implementierten Übertragungsmatrizen $\underline{A}_i, \underline{L}_i$ nach [RUB93] enthalten auch die Einflüsse der Theorie II-ter Ordnung und einer elastischen Bettung. Am Ende des Feldes wird aus der Übertragungsmatrix eine Steifigkeitsmatrix (Koppelfeder) erstellt, die beim nachfolgenden Feld berücksichtigt wird. Grundlage des Rechenprozesses ist die Klasse *ZUST* der Zustandsvektoren. Durch sukzessive Anwendung der Methoden *Aufstellen*, *Koppelfeder_anschliessen*, *übertrage_über_Abschnitt*, *übertragen_über_Punkt* und *bestimme_Koppelfeder* der Klasse *ZUST* wird das Übertragungsverfahren im objektorientierten Entwurf abgebildet. Bild 1 erläutert den Ablauf anhand eines Sequenzdiagramms in UML-Notation.

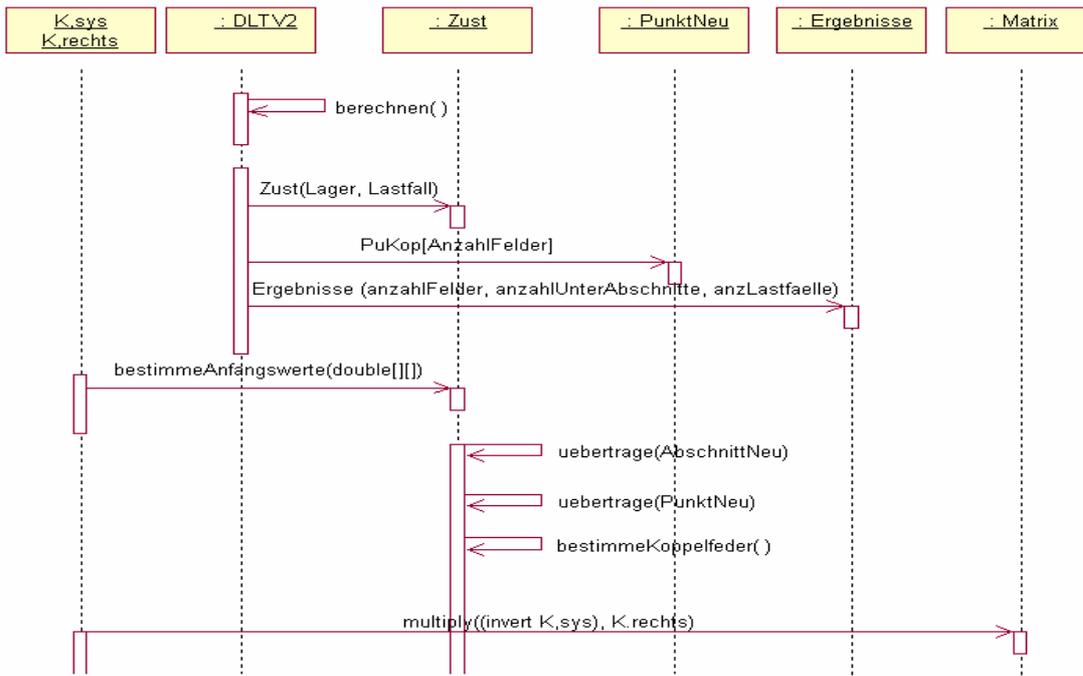


Bild 1: Sequenzdiagramm der Methode *berechnen* der Klasse DLTv2

3 Anwendungsprogrammierung

Die im Framework implementierten COM-Klassen können als Server von unterschiedlichen Programmen genutzt werden. Für Engineering Desktop Anwendungen sind Office- und CA-Software geeignet. Bei EDATRA wurden MS-Excel und das Programm Mathcad der Firma MathSoft verwendet. Ein Überblick über das Konzept von EDATRA gibt Bild 2. Die Clientsoftware greift über ihre Programmierschnittstelle auf die COM-DLL des Frameworks zu. Bei den MS-Office-Produkten handelt es sich hierbei um die Programmiersprache VBA (*Visual Basic for Applications*), bei Mathcad ist der Weg über eine Funktionsprogrammierung mit C++ oder mit einer Skriptingsprache (VBScript und JScript) möglich. Die Client-Programmierung mit VBA und VBScript bzw. JSkript ist vergleichsweise einfach und kann durchaus auch von einem erfahrenen Anwender durchgeführt werden.

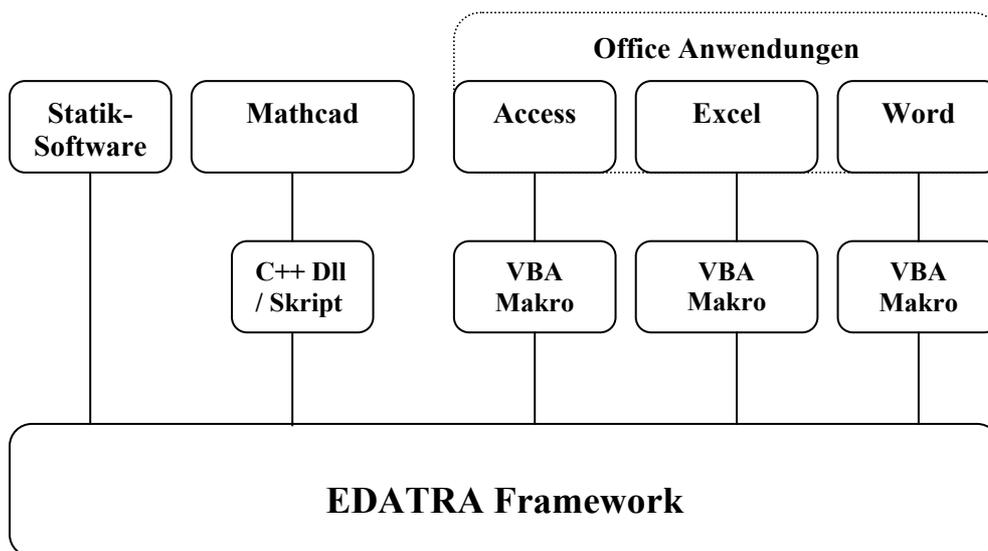


Bild 2: Framework für die Tragwerksplanung

Darüber hinaus lässt sich das EDATRA-Framework auch für eigenständige Programme nutzen. Neben den Anwendungen in MS-Excel und Mathcad wurden als Pilotprojekte eine Anwendung in MS-Access, eine eigenständige Anwendung in VB (*Visual Basic*) sowie eine Internetanbindung des Frameworks für Stabwerke entwickelt.

4 Engineering Desktop Anwendungen des Frameworks

4.1 Entwurfsstatik

Im frühen Stadium der Tragwerksplanung oder bei der überschlägigen Prüfung von Computerberechnungen wird heute durchaus noch „Papier und Bleistift“ dem Computer als Werkzeug in der Praxis vorgezogen. Werkzeuge wie Mathcad können aber die Arbeit wesentlich vereinfachen und beschleunigen, insbesondere, wenn geeignete Tools für die Tragwerksplanung zur Verfügung stehen [WER03].

Bild 3 zeigt den Ausschnitt einer Anwendung in Mathcad. Im Rahmen einer statischen Voruntersuchung wurden die Momentengrenzlinien mit den EDATRA-Funktionen des Durchlaufträgers bestimmt. Die an der Stelle des maximalen Moments erforderliche Bewehrung wird mit der Funktion *SB_BemB* (Stahlbeton_Bemessung Biegung) ermittelt. Alternativ kann ein Stahlprofil für Biegeknicken und Biegedrillknicken untersucht werden.

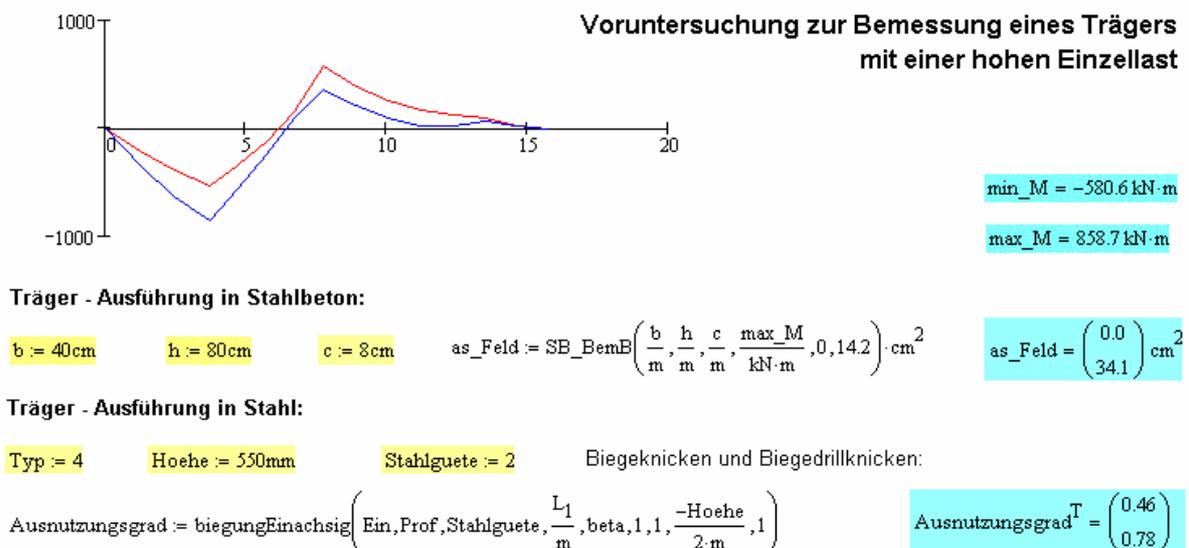


Bild 3: Anwendung des EDATRA-Frameworks bei Funktionen in Mathcad

4.2 Teachware

Mit Hilfe geeigneter Teachware können statische Zusammenhänge den Studierenden im Sinne eines numerischen Experiments lebendig und anschaulich vermittelt werden. Das Edatra-Framework unterstützt die Entwicklung von Teachware wesentlich. Bild 4 zeigt ein Beispiel in Excel, bei dem zunächst die Knicklast einer Stütze bestimmt und damit der Einfluss nach Theorie II-ter Ordnung mit Hilfe der Näherungsformel von Dischinger ermittelt wird. Zum Vergleich wird auch eine exakte Berechnung nach Theorie II-ter Ordnung durchgeführt. Für alle statischen Berechnungen wird das EDATRA-Framework genutzt.

System:

	Feld 1:	Feld 2:
Länge L [m]	6	5
Steifigkeit EI [kN m ²]	8000	8000
Last q [kN/m]	0	2
Normalkraft N [kN]	300	100

Verzweigungslastfaktor N_{ki}/N

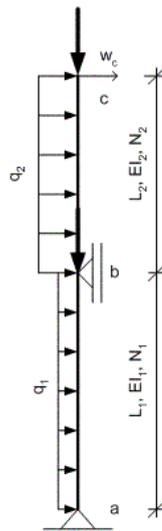
Lastfaktor der Normalkräfte:	3,30
Durchbiegung w_c [m]:	76,77
Normalkräfte N [kN]:	990 330

Stützmoment M_b

Theorie II-ter Ordnung	-32,19 [kNm]
Theorie I-ter Ordnung	-25,00 [kNm]

Dischinger-Formel

M_b nach Dischinger-Formel:	-35,87 [kNm]
$M_{Dischinger}/M_{II}$:	1,11



Dischinger-Formel

Vergrößerungsfaktor der Schnittgrößen nach Theorie II-ter Ordnung

$$S_{II} = \frac{1}{1 - \frac{N}{N_{kI}}} \cdot S_I$$

S_I Schnittgröße nach Theorie I-ter Ordnung
 S_{II} Schnittgröße nach Theorie II-ter Ordnung
 N_{kI}/N Verzweigungslastfaktor

Literatur:
 Petersen, Statik und Stabilität der Baukonstruktionen, Vieweg, 1992, Seite 168

Bild 4: Anwendung des EDATRA-Frameworks bei einem Teachtoll in Excel

5 Zusammenfassung und Ausblick

Frameworks für die Tragwerksplanung stellen ein effizientes Werkzeug zur Entwicklung unterschiedlichster Anwendungssoftware dar. Die Nutzeroberfläche kann nicht nur aus Berechnungssoftware wie Excel und Mathcad bestehen, auch die Entwicklung von Tragwerkseditoren ([BIT00]) oder internetbasierender Software ist beispielsweise damit möglich. Ein Ziel könnte die Standardisierung von Klassen und Methoden in der Tragwerksplanung sein, um damit eine Wiederverwendbarkeit von Anwendungssoftware auch bei einem Wechsel des Frameworks sicherzustellen.

Literatur

- [BIT00] Bittrich D., Verbunddokumente als Nutzeroberfläche von Software für die Tragwerksplanung, IKM 2000, Bauhaus Universität Weimar 2000
- [GRU00] Gruhn, V, Komponentenmodelle, Addison-Wesley 2000
- [HOF01] Hofmann, J., Jobst, F., Schabenberger R., Programmieren mit COM und CORBA, Hanser 2001
- [RUB93] Rubin H., Baustatik ebener Stabwerke, Stahlbau Handbuch Band 1 Teil A, 3. Aufl., 1993, Stahlbau Verlags GmbH, Köln
- [WER02] Werkle H., H. Pleßke, ED_TRA Engineering Desktop Anwendungen für die Tragwerksplanung, F&E-Bericht, Institut für angewandte Forschung, Fachhochschule Konstanz, 2002
- [WER03] Werkle H., R. Avak, Mathcad in der Tragwerksplanung, Vieweg, Wiesbaden 2003