

## Sklavenaufstand



Tilman Börner  
Chefredakteur dotnetpro

**G**lauben Sie wirklich, der Rechner wäre Ihr „Knecht“? Ich will Sie ja nicht schocken, aber ich behaupte: Es ist genau andersherum. Sie arbeiten, während der Computer die meiste Zeit auf der faulen Haut liegt. Rauchende Köpfe und schmerzende Finger vor der Mattscheibe, Faulenzen und dolce far niente dahinter. So schaut es aus.

Dabei gilt, so lange ich in der Computerei bin, dass ein Rechner gar nicht schnell genug sein kann. Ein Widerspruch?

Immer noch schneller mussten die Prozessoren sein. Hier noch fünf Prozent Performance herausgequetscht, nur damit der Benchmark drei Sekunden früher fertig ist. Doch: Bis auf 3-D-Rendering und Spiele gibt es kaum Anwendungen, die den Prozessor so richtig ins Schwitzen bringen. Was soll also die Forderung nach dem schnellen Rechner?

Bitte verstehen Sie mich nicht falsch: Lahme Computer sind mir ein Graus. Es ist meines Erachtens menschenunwürdig, auf so eine graue Kiste zu warten. Und seien es nur Sekunden, die sie braucht, bis sie eine Webseite darstellt. Die Kiste raubt mir Lebenszeit. Immer wieder kleine Stückchen. Dabei faulenzet der Schlawiner die meiste Zeit, denn in den seltensten Fällen liegt es am Prozessor, wenn ich warten muss. Die Flaschenhalse liegen meist woanders.

Was also tun mit der vorhandenen Rechen-Power? Schnellere Rechner reduzieren die Turnaround-Zeiten, weil der Compiler schneller den Code ausspuckt. Guter Punkt. Das ist eine konkrete Hilfe für uns Programmierer.

Ich habe noch einen anderen Vorschlag: Lassen wir den Kerl doch zur Laufzeit ackern und machen uns die Arbeit zur Designzeit leichter, in-

dem wir – saubere Schnittstellen definieren und einsetzen. Was hat das jetzt mit der Geschwindigkeit zu tun, werden Sie fragen. Nun, sehr viel. Würden wir jede Klasse nur über Eigenschaften und Methoden und nicht über Felder, die als public deklariert sind, zugänglich machen, wäre eine Anwendung stabiler aufzubauen. Das würde auf der anderen Seite aber auch mehr Arbeit für den Rechner bedeuten.

Gäbe es genügend schnelle Rechner, könnten wir endlich das Schichtendesign sauber umsetzen und müssten nicht aus Performance-Gründen an bestimmten Stellen um ein sauberes Interface herum programmieren.

Nehmen wir also endlich die Rechen-Power, die uns zur Verfügung steht, und machen wir damit, was uns Programmierern zugute kommt. dotnetpro unterstützt mit dem Schwerpunkt Patterns (ab Seite 12) diesen Appell. Mit den passenden Mustern lässt sich der Code besser strukturieren und wird lesbarer. Hier gibt es eine ganze Reihe an Mustern, deren Einsatz quasi in jeder Anwendung dienlich ist.

Neben dem Schichtendesign liegen verteilte Anwendungen und SOA im Trend. Ohne Schnittstellen laufen solche Anwendungen nicht, da die Einzelteile nicht miteinander kommunizieren können. Torsten Zimmermann zeigt ab Seite 118, wie Sie Microkernel-Strukturen in Ihren Anwendungen sinnvoll einsetzen können. Und Achim Oellers macht ab Seite 98 klar, was denn nun SOA tatsächlich ist.

Stehen Sie also auf. Emanzipieren Sie sich. Befreien Sie sich von den Fesseln und fordern Sie Ihren Rechenknecht. Lassen Sie ihn arbeiten. dotnetpro zeigt Ihnen, wie Sie das anstellen.

Viel Spaß beim Lesen wünscht Ihnen

Tilman Börner  
Chefredakteur dotnetpro