

A. Schindler: „Abstraktion führt zu Produktivitäts- und Leistungsschüben um Größenordnungen“

Unter Entwicklern ist der Glaube weit verbreitet, Moores Gesetz, welches besagt, dass sich die Transistordichte eines Chips alle 18 Monate verdoppelt, gebe eine Obergrenze für die Geschwindigkeit an, mit der sich die Systemleistung erhöhen kann. Die Chipdichte stellt aber lediglich ein Sprungbrett für jene architektonischen und algorithmischen Verbesserungen dar, die bahnbrechende Leistungssprünge ermöglichen. Jeder dieser Sprünge wurde erst durch neue Entwicklungswerkzeuge ermöglicht, die auf die Kernprobleme ihrer jeweiligen Zeit zugeschnitten waren und eine um Größenordnungen gesteigerte Produktivität ermöglichten. Entwickler sehen sich außerdem der Forderung ausgesetzt, zunehmend komplexere Systeme in immer kürzeren Zeiträumen abzuliefern. Einige Beobachter der Branche haben schon auf eine drohende ‚Softwarekrise‘ oder ‚Produktivitätskrise‘ hingewiesen.



ANDREAS SCHINDLER,
Geschäftsführer,
The MathWorks GmbH

Herkömmliche hardware- und softwareorientierte Sprachen bieten nicht die Abstraktion, die nötig ist, um algorithmisch anspruchsvolle Systeme effizient zu entwickeln. Umgebungen für die Systementwicklung müssen es dem Entwickler erlauben, sowohl die algorithmische als auch die architektonische Komplexität exakt wiederzugeben. Um komplexe Systeme erfolgreich entwickeln zu können, ist eine Entwicklungsumgebung gefordert, in der das gesamte System einschließlich sämtlicher Wechselwirkungen der hardware- und der softwareseitigen Subsysteme modelliert werden kann. ▶

Genau dies ist die Grundlage für einen modellbasierten Entwicklungsansatz in der Entwicklung, der mit MATLAB/Simulink umgesetzt werden kann: Kontinuierliches Simulieren, Verfeinern und Testen des Systemmodells von der idealisierten Spezifikation bis hin zu einer bitgenauen und das Timing exakten wiedergebenden Darstellung, automatische Code-/Systemgenerierung und die Möglichkeit, ältere Technologien zu integrieren. Fortschritte in der damit möglichen Abstraktion und der Code-/Systemgenerierung führen unausweichlich zu Produktivitäts- und Leistungsschüben um ganze Größenordnungen. Diese Herangehensweise beseitigt Fehler früher im Entwicklungsprozess und verringert somit das Risiko von Fehlentwicklungen und verfehlten Terminen. Außerdem bie-

tet sie den Vorteil, nicht auf eine einzige konkrete Implementierung festgelegt zu sein. Neue System-Plattformen können durch gezielte Modifikationen nur an den notwendigen Stellen im System voll genutzt werden. Es mag einige Zeit dauern, bis sich solche neuen Entwicklungsmethoden durchsetzen. Aber nur mit der zielgerichteten Einführung dieser Techniken lassen sich langfristig die Aufgaben bei der Entwicklung von Hard- und Softwaresystemen in einem globalen Wettbewerb erfolgreich meistern. ■

Beitrag als PDF auf www.duv24.net

