

## R. Kaltenbrunner: „Die Zukunft der Entwicklung von Embedded-Softwaresystemen“

Der Markterfolg und die Innovation zahlreicher Produkte hängen immer mehr von der Weiterentwicklung der integrierten Mikro-



ROLAND KALTENBRUNNER  
ist Geschäftsführer der  
Easycode GmbH

prozessoren ab. In der Embedded-Entwicklung steigen dadurch zunehmend die Anforderungen und die Komplexität.

- ▶ In modernen Mittelklasseautos sind nicht selten insgesamt bis zu 50 verschiedene Mikroprozessoren integriert.
- ▶ In der Telekommunikation enthält jedes Handy sowie Mobilteil für Festnetztelefone oder jeder komplexe Vermittlungsrechner eine Vielzahl von Mikro- bzw. Multiprozessoren.
- ▶ In der Luft- und Raumfahrt kommt umfang-

reich eingebettete Intelligenz zum Einsatz.

- ▶ In der Produktionsautomatisierung werden verteilte Steuerungen und einzelne Sensoren mit immer mehr Rechenleistung ausgestattet.

Mit der stets steigenden Komplexität kommt auch Bewegung in die Methode der Softwareentwicklung. Gerade die klassische Art der Embedded-Entwicklung ist nach wie vor durch Brüche zwischen den einzelnen Projektschritten, verursacht durch die Definition in Form von Text, Pseudocode und verschiedenen Diagrammen, geprägt. Oft werden die Ergebnisse einer Phase neu interpretiert und durch den jeweiligen Entwickler in eine andere Darstellung gebracht. Dieser Problematik wird durch bereits auf dem Markt befindliche Tool-Sets wie zum Beispiel ‚\_Vision‘ von Keil in Zusammenspiel mit EasyCODE C/C++ Rechnung getragen.

Die bereits heute in der Anwendungsentwicklung als Standard geltende UML, gewinnt zunehmend auch in der Embedded-Entwicklung an Bedeutung. Auf den ersten Blick kennt die UML-Spezifikation kaum Diagrammtypen, die für die Embedded-Entwicklung von Nutzen sind. Wie sollen Use-Case- oder Activity-Diagramme sinnvoll in der Embedded-Entwicklung verwendet werden? In der Real-Time-Umgebung sind eher Darstellungsformen wie Practical State Charts

(nach Miro Samek) von Nutzen. Die Weiterentwicklung der UML berücksichtigt allerdings mehr und mehr auch diese Anforderungen.

In der Praxis kommt schon jetzt eine zwar ungewöhnliche aber bewährte Methode zum Einsatz, die UML mit Struktogrammen verbindet. Durch die Nutzung der Vorteile der jeweiligen Methoden, einerseits UML und andererseits der Struktogramme nach Nassi Shneiderman, erreicht man die Konsistenz zwischen Modell, Code und Dokumentation. Der Quellcode, der durch den Entwurf im UML-Tool entsteht, kann integriert im Struktogramm dargestellt und in den späteren Projektphasen, wenn die Kräfte der UML zunehmend ausgezehrt sind und der Entwickler normalerweise alleine gelassen wird, entwickelt und automatisch dokumentiert werden. Eine der Lösungen für diese Vorgehensweise liegt mit Objectif von Microtool und EasyCODE C++ vor.

Die Notwendigkeit der Verteilung von Embedded-Software auf mehrere Embedded Targets, die in statischen oder dynamischen Netzwerken zusammengefasst werden, schafft ganz neue Herausforderungen an die Softwareentwicklung. Die Softwarearchitektur die hier zur Anwendung kommt, stellt völlig andere Anforderungen an die Tool-Landschaften und erfordert bzw. garantiert somit eine kontinuierliche Weiterentwicklung der Methoden und Tools in diesem Bereich. ■

Beitrag als PDF auf [www.duv24.net](http://www.duv24.net)

more @ click

TG0410

