

Komponentenbasierte Softwareentwicklung für eingebettete Echtzeitsysteme

Uwe Rasthofer

Universität Erlangen-Nürnberg
Lehrstuhl für Betriebssysteme

Januar 2000

Viele Komponentenmodelle, wie Microsoft COM und Java Beans, basieren darauf, daß eine uniforme Ausführungsplattform und nahezu unbegrenzte Ressourcen zur Verfügung stehen. Es wird angenommen, daß sich die Komponenten nur über ihre (funktionalen) Schnittstellen beeinflussen. Nichtfunktionale Eigenschaften werden nicht betrachtet. Ein Komponentenmodell für eingebettete Echtzeitsysteme muß jedoch nichtfunktionale Eigenschaften, wie Laufzeiten und Speicherverbrauch, mit berücksichtigen, da diese die Korrektheit des Systems wesentlich bestimmen (siehe harte Echtzeitsysteme). Die nichtfunktionalen Eigenschaften der einzelnen Komponenten und des Gesamtsystems sind jedoch von der Ausführungsplattform abhängig, was zu einer geringen Wiederverwendbarkeit und Anpaßbarkeit führt.

Hier wird ein Komponentenmodell vorgestellt, das die Funktionalität der Komponenten von nichtfunktionalen Aspekten trennt (siehe Aspect-Oriented Programming). Die Komponenten sind ereignisgesteuert und ihr Verhalten wird zunächst unabhängig von einer Ausführungsplattform von Programmierer beschrieben. Aus diesen plattformunabhängigen Komponenten wird dann die Anwendung aufgebaut. Der Anwendungsentwickler kann nun zusätzliche Anforderungen an das Verhalten des Gesamtsystems in Form von Zeitbedingungen an den Eingangs- und Ausgangsereignissen definieren. Die Verbindung zwischen zusammengehörenden Eingangs- und Ausgangsereignissen erfolgt durch kritische Pfade, die durch das Gesamtsystem gelegt werden.

Erst nachdem die Funktionalität und die Zeitbedingungen des Gesamtsystems feststehen, wird die Ausführungsplattform festgelegt. Mit Hilfe einer Beschreibung der Plattform oder durch Messungen können Laufzeiten und Speicherverbrauch bestimmt werden. Anhand des daraus abgeleiteten Ablaufplanes werden Komponenten identifiziert, die nebenläufig ausgeführt werden und daher koordiniert werden müssen. Diese Komponenten werden mit Synchronisationscode ausgestattet und der Ablaufplan wird entsprechend angepaßt. An dem so entstandenen System lassen sich nun die Zeitbedingungen der Anwendung überprüfen.

Das hier vorgestellte Modell erlaubt eine einfache Wiederverwendung der Komponenten und eine Anpassung an verschiedene Ausführungsplattformen unter Beachtung nichtfunktionaler Eigenschaften. Aufgrund der Kommunikation zwischen den Komponenten über Ereignisse ist auch eine Erweiterung für verteilte Anwendungen einfach möglich.