

# Systemerstellung unter dem Echtzeitbetriebssystem Ambrosia/MP

Dipl.Inform. Ingo Stierand

Universität Oldenburg  
Abteilung Betriebssystem und Verteilte Systeme

Januar 2000

Im Vergleich zu klassischen Rechensystemen hat im Bereich der eingebetteten Systeme die möglichst optimale Nutzung der vorhandenen Hardware-Ressourcen einen hohen Stellenwert. Neben dem Kostenfaktor spielen hier oft räumliche Beschränkungen eine Rolle. In größeren Systemen, wie sie beispielsweise im Automobilbereich vorzufinden sind, zeigt sich ein eindeutiger Trend zu vernetzten und verteilten Rechensystemen. In dem Projekt Ambrosia/MP wird das gleichnamige Echtzeitbetriebssystem entwickelt, mit dem eine effiziente Ressourcennutzung in verteilten, eingebetteten Systemen möglich ist.

Ambrosia/MP zeichnet sich durch die Fähigkeit aus, ausgehend von einem Basissystem anwendungsangepasste Laufzeitsysteme zur Verfügung zu stellen. Dabei werden bereits zur Entwicklungszeit mit Hilfe von sogenannten Komponenten die verwendeten Systemdienste vollständig an die Hardware und an die Anforderungen der Anwendung angepasst. Die entstehenden Laufzeitsysteme enthalten dann genau die für die Ausführung relevanten Elemente. Die Auswahl und Instanziierung der Komponenten basiert zum Einen auf der zugrundeliegenden Rechnerstruktur und zum Anderen auf den durch den Anwendungsentwickler mittels entsprechender Werkzeuge eingegebenen Konfigurationsinformationen.

Dieses Komponentenmodell lässt eine hohe Komplexität bei der Erstellung der Laufzeitsysteme, insbesondere in Systemen mit mehreren Recheneinheiten, erkennen. Die Umsetzung der Struktur- und Konfigurationsinformationen in ein lauffähiges System erfordert ein komplexes Regelwerk, das innerhalb eines Übersetzungswerkzeuges kodiert sein muss. An dieser Stelle setzt die im Rahmen des Projektes entwickelte Konfigurationssprache *AMScribe* an. Das dabei umgesetzte Sprachkonzept verfolgt mehrere Ziele. Um zu vermeiden, dass bei Änderungen bzw. Erweiterungen der Betriebssystembasis immer auch die Werkzeuge anzupassen sind, kann das gesamte zur Umsetzung notwendige Regelwerk innerhalb der Sprache ausgedrückt werden. Ein generischer, als Bibliotheksmodul in das Werkzeug integrierter Übersetzer wird mit dem Regelwerk instanziiert und übernimmt dann die Systemerzeugung. Um die Interoperabilität zu anderen Entwicklungswerkzeugen, wie beispielsweise Case-Tools oder Spezifikationswerkzeugen zu vereinfachen, werden alle zur Systemerstellung notwendigen Parameter ebenfalls in der Sprache ausgedrückt, so dass ein einfacher Import- bzw. Export von Konfigurationsinformationen möglich ist. Desweiteren unterstützt das Sprachkonzept die grafische Darstellung und Manipulation der Konfigurationsparameter. Als Bibliotheksmodul organisiert stehen entsprechende Dialogfelder zur Verfügung, mit denen die Entwicklung von Konfigurationswerkzeugen wesentlich vereinfacht wird.