

Zulieferung und Integration in Automotive Produktlinien: Konfiguration von Prozessreihenfolgen innerhalb von Tasks

Birgit Boss

Robert Bosch GmbH
P.O. Box 30 02 40
D-70442 Stuttgart
Birgit.Boss@de.bosch.com

Abstract: Durch die steigende Komplexität von Steuergeräte-Software, die Wiederverwendung von SW-Komponenten in verschiedenen Produkten einer Produktfamilie sowie den steigenden Anteil zugelieferter Software verstärkt sich die Anforderung, die Konfiguration der zeitlichen und sequentiellen Ausführung von Prozessen in Tasks systematisch und formal je SW-Komponente zu erfassen, so dass in den Projekten eine werkzeugunterstützte Überprüfung stattfinden kann. Es wird dabei ein OSEK-konformes statisches Betriebssystem vorausgesetzt. Das entwickelte und pilotierte Konzept setzt auf dem MSR-Backbone-Gedanken auf: die durchgängige Verwendung eines xml-basierten standardisierten Austauschformats für die komplette Entwicklung und Zulieferung. Dabei wurde speziell auf den Aspekt der Variationspunkte Wert gelegt, der in Produktlinien wichtig ist.

1 Einleitung

Im Folgenden wird ein OSEK-konformes statisches Betriebssystem vorausgesetzt. OSEK-OS ist eine ISO-zertifizierte (17356-3) Spezifikation für Betriebssysteme eingebetteter Systeme [OSEK]. Die einzigen ausführbaren Objekte in OSEK-OS sind Tasks. Der sequentiell auszuführende Code wird über die Reihenfolge der Prozesse in den Tasks festgelegt. Prozesse entsprechen Threads oder Runnables. Der OSEK-OS Standard ist Basis in AUTOSAR [AUTO]. Wichtiger Input für die AUTOSAR-SW-Komponenten-Beschreibung ist die Dokumentationsspezifikation MSRSW des Standards ASAM AAS (ASAM Adopted Specifications) [ASAM]. ASAM und AUTOSAR sind Standardisierungs-Initiativen der Automobilindustrie mit dem Fokus auf SW-Austausch. Aktuell wird das Thema Prozessreihenfolgen dort nicht betrachtet.

Das geforderte sequentielle Ausführungsverhalten innerhalb der Tasks (= Prozessreihenfolgenvorgabe) von SW-Komponenten wurde bislang textuell beschrieben, z. B. in Integrationsanleitungen. Zum Teil lag die Information verstreut vor. Durch die steigende Komplexität von Steuergeräte-Software, die Wiederverwendung von SW-Komponenten

in verschiedenen Produkten einer Produktfamilie sowie den steigenden Anteil zuge-
lieferter Software kam jedoch verstärkt die Anforderung auf, auch diesen Teil der
Schnittstelle einer SW-Komponente systematisch und formal zu erfassen, so dass in den
Projekten eine werkzeugunterstützte Überprüfung stattfinden kann. In [Ti03] wird be-
schrieben, wie MSRSW als XML-Backbone für die gesamte Produktlinien-Entwicklung
sowie den Software-Austausch zwischen OEM und Zulieferer verwendet werden kann.
MSRSW ist die Grundlage für das hier verwendete und vorgestellte Austauschformat.

2 Konfiguration von Prozessreihenfolgen

Konfiguration und Variationspunkte. Die Konfiguration eines Produkts in Produkt-
linien erfolgt über zwei Schritte:

- (1) Auswahl der einzubindenden SW-Komponenten in ihrer jeweiligen Alternative und
Version (geschieht i. d. R. über das Konfigurations-Management-System).
- (2) Setzen der Variationspunkte, die in den aus Schritt 1 gewählten SW-Komponenten
noch offen sind (geschieht i. d. R. über bedingte Kompilierung).

Der in diesem Artikel vorgestellte Ansatz verwendet in beiden Konfigurationsschritten
denselben Mechanismus zur Spezifikation der Variabilitäten. In MSRSW geschieht dies
durch konsequente Verwendung des XML-Tag <SW-SYSCOND>: ein Variationspunkt
wird durch eine logische Bedingung ausgedrückt¹. Die Prädikate in einer solchen Be-
dingung entsprechen so genannten Systemkonstanten. Diese Systemkonstanten werden
von der Integration produktspezifisch gesetzt und so ergibt sich aus den bedingten Vor-
gaben eine eindeutige Vorgabe für das Produkt.

Prozessreihenfolge einer SW-Komponente. Die Prozessreihenfolgenvorgabe gehört
zur Schnittstelle einer SW-Komponente. Durch die Zuordnung von Prozessen zu Tasks
wird das zeitliche Verhalten der SW-Komponente festgelegt². Ein Prozess darf in mehr
als einer Task laufen. Als Beispiel betrachten wir die Prozessreihenfolgenvorgabe der
Alternative FFC_1 der SW-Komponente FFC für die Task „OS_20ms_Task“ in
Abbildung 1 (A.1 und A.2).

Es handelt sich um eine Task, die alle 20ms ausgeführt wird. Nehmen wir an die
Systemkonstante SY_BDG sei auf 1 oder 2 gesetzt. Dann folgt daraus, dass folgende
Prozessreihenfolge die geforderte ist: FFC_A_Proc → FFC_B_Proc → FFC_C_Proc
→ FFC_D_Proc. Ist dagegen SY_BDG gleich 0, so ändert sich die geforderte Reihen-
folge zu FFC_A_Proc, → FFC_D_Proc → FFC_B_Proc → FFC_C_Proc.

¹ Variationspunkte gehören zur Spezifikation der Schnittstelle und werden deshalb explizit modelliert und nicht
durch XML-Verarbeitungsanweisungen (Processing Instructions) realisiert.

² Das Task-Scheduling wird in der dynamischen Sicht der Produktlinien-Architektur festgelegt [St04].

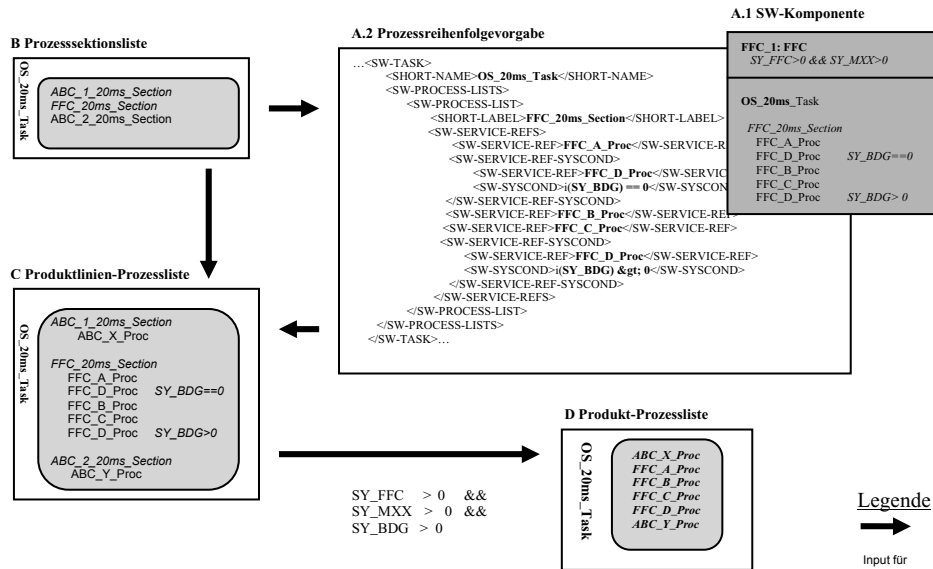


Abbildung 1: Prozessreihenfolgen in der SW-Komponente, der Produktlinie und dem Produkt³

Alternativen einer SW-Komponente. Die Definition und Deklaration von Prozessen erfolgt unabhängig von der Spezifikation der Prozessreihenfolge und der Zuordnung zu Tasks in einem früheren Entwurfsschritt. Verschiedene alternative Implementierungen einer SW-Komponente können unterschiedlich viele Prozesse definieren. Auch die Reihenfolge von Prozessen sowie die Zuordnung zu Tasks kann sich je nach Alternative oder Version ändern. Dies sollte allerdings vermieden werden. Betrachten wir wieder das Beispiel aus Abbildung 1: Die SW-Komponente FFC darf nur genau dann in ein Projekt eingebunden werden, falls $SY_FCC > 0$. Alternative FFC_1 setzt voraus, dass $SY_MXX > 0$ (A.1). Die zweite (nicht gezeigte) Alternative FFC_2 dagegen fordert $SY_MXX == 0$.

Prozessreihenfolgen zwischen SW-Komponenten. Es ist nicht ausreichend, wenn jede einzelne SW-Komponente Prozessreihenfolgevorgaben spezifiziert: im Projekt wird festgelegt, in welcher Reihenfolge die Prozesse aller integrierten SW-Komponenten ablaufen werden. Als Unterstützung für diese Aufgabe wird seitens der SW-Architekten je Task und SW-Komponente eine Menge von so genannten Prozesssektionen definiert und eine Reihenfolge dieser Prozesssektionen innerhalb der Task festgelegt. Die so entstehende Liste nennen wir Prozesssektionsliste. Die SW-Komponente muss bei ihrer Schnittstellenvorgabe jedem ihrer Prozesse eine Prozesssektion zuweisen. So wird die Reihenfolge eindeutig. Für die Produktlinie in Abbildung 1 liegt folgende Prozesssekti-

³ i(SY_BDG) ist die Kurzschreibweise zu <SW-SYSTEMCONST-CODED-REF>SY_BDG</SW-SYSTEMCONST-CODED-REF>. Es handelt sich um eine Referenz auf eine Systemkonstante <SW-SYSTEMCONST> namens SY_BDG. Die Systemkonstante wird in einem früheren Entwurfsschritt definiert.

onsliste für die 20ms-Task vor (B): ABC_1_20ms_Section → FFC_20ms_Section → ABC_2_20ms_Section. Die SW-Komponente ABC hat dabei 2 Prozesssektionen zugeordnet bekommen. Dazwischen laufen Prozesse der SW-Komponente FFC.

Ableitung von Produkten aus der Produktlinien-Prozessliste. Im Gegensatz zur Prozessliste eines Produkts zeichnet sich die Prozessliste einer Produktlinie, die als Basis für die Ableitung der Produkte verwendet werden kann, dadurch aus, dass noch Variabilitäten enthalten sind. Abbildung 1 zeigt die Prozessliste der Produktlinie (C) sowie die Prozessliste eines daraus abgeleiteten Produkts (D).

3 Zusammenfassung und Ausblick

In diesem Artikel wurde ein XML-Format für die Festlegung und Handhabung von Prozessreihenfolgen in Produktlinien vorgestellt. Das Konzept und ein entwickeltes prototypisches Werkzeug „ProcMan“ zur Auswertung der Vorgaben in Projekten werden aktuell pilotiert. Die Akzeptanz bei den Anwendern ist aufgrund des hohen Verbesserungspotenzials groß. Da die Abdeckung bzgl. der Vorgaben noch nicht ausreichend ist, wurde das Konzept dadurch erweitert, dass zusätzlich zu den Vorgaben Lese-/Schreibbeziehungen von globalen Variablen mit in die Analyse einbezogen werden. Diese Analyseergebnisse sind außerdem ein Hilfsmittel bei der Qualitätsabsicherung der Prozessreihenfolgenvorgaben.

4 Literaturverzeichnis

- [ASAM] ASAM – Association for Standardisation of Automation and Measuring Systems. <http://www.asam.net/> zuletzt besucht am 31.01.2007.
- [AUTO] AUTOSAR – Automotive Open System Architecture. <http://www.autosar.org/> zuletzt besucht am 31.01.2007.
- [OSEK] OSEK/VDX - Offene Systeme und deren Schnittstellen für die Elektronik im Kraftfahrzeug. <http://www.osek-vdx.org/> zuletzt besucht am 31.01.2007.
- [St04] Steger, M. et. al.: Introducing PLA at Bosch Gasoline Systems: Experiences and Practices. In: Proc. of Software Product Lines Conference (SPLC 2004), Boston, 2004; Springer Lecture Notes.
- [Ti03] Tischer, C. et. al: Offene Software Systeme basierend auf der EDC/ME(D)17 Architektur. VDI-Kongress „Elektronik im Kraftfahrzeug“, Baden Baden, 2003.