

Adaptierbare Software-Architektur für den Software-Download in Kfz-Steuergeräte

Cornelia Heinisch

STZ Softwaretechnik

Cornelia.Heinisch@stz-softwaretechnik.de

Martin Simons

DaimlerChrysler AG

Martin.Simons@daimlerchrysler.com

Abstract: Zukünftig sollen die Steuergeräte eines Fahrzeugs in der Entwicklung, in der Fertigung und jederzeit nach der Auslieferung mit neuer Software versorgt werden können. Eine Funktionsaktualisierung oder eine Fehlerbeseitigung in der Steuergeräte-Software kann dann auch durch einen Softwaretausch erfolgen und ist nicht mehr auf den Austausch von Hardware beschränkt. In diesem Beitrag wird eine Software-Architektur für den Software-Download in Fahrzeug-Steuergeräte vorgestellt, die auf einfache Weise an die unterschiedlichen Rahmenbedingungen in Entwicklung, Fertigung und Außenorganisation adaptiert werden kann. Die Software-Architektur und ein Prototyp zur Evaluierung der Architektur wird in diesem Beitrag vorgestellt.

1 Einleitung

Unter Software-Download in ein Kraftfahrzeug wird sowohl das Installieren zusätzlicher Software als auch die Aktualisierung bestehender Software – sei es zur Fehlerbeseitigung oder zur Anpassung der Funktion an den Stand der Technik – verstanden. Der Software-Download wird künftig in allen Fahrzeugdomänen (Chassis, Motorraum, Komfort & Karosserie und Telematik) zum Einsatz kommen, um das steigende Software-Volumen im Kraftfahrzeug wartbar zu machen [HS03], [SC03]. Jedes Steuergerät, in das neue Software geladen werden kann, besitzt einen Flashspeicher. Dieser Flashspeicher wird bei einem Software-Download zuerst gelöscht und dann mit der neuen Software wieder beschrieben. Den Vorgang, neue Software in ein Steuergerät zu laden, nennt man Flashen und die Software, die in das Steuergerät geladen wird, ist die sogenannte Flashware.

Es ist zu beachten, dass ein Software-Download in Steuergeräte sowohl in der Entwicklung, bei der Fertigung am Band als auch in der Außenorganisation möglich sein muss. Bei der Erstprogrammierung der Steuergeräte während der Produktion eines Fahrzeugs hat man hierbei die besonders kritische Anforderung, alle flashbaren Steuergeräte eines Fahrzeugs in möglichst kurzer Zeit am besten parallel zur Produktion zu flashen. Eine Software-Architektur, die den Download-Vorgang in Entwicklung, Fertigung und Außenorganisation steuern und überwachen soll, muss entsprechend Steuergeräte in beliebigen Fahrzeugdomänen reprogrammieren (flashen) können und überdies den Bezug neuer Software (Flashware) über verschiedene Übertragungswege und -medien erlauben (siehe Bild 1-1).

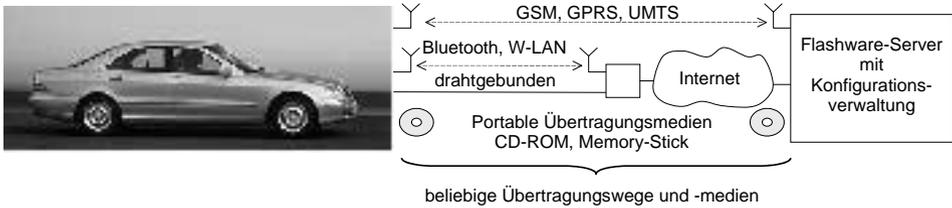


Bild 1-1: Software-Download in das Fahrzeug über beliebige Übertragungswege und -medien

Die zu ladende Software (Flashware), die auf einem Flashware-Server gelagert wird, kann beispielsweise über folgende Wege und Medien in das Fahrzeug gelangen:

- drahtlos über GSM, GPRS, UMTS oder über Broadcast-Netze wie DAB von der Infrastruktur,
- über Bluetooth, Wireless-LAN oder drahtgebunden in der Werkstätte, an Software-Tankstellen oder am Band,
- über portable Übertragungsmedien wie CD-ROM oder USB Memory-Stick.

Um die Software für die Entwicklung, für die Fertigung am Band und in der Außenorganisation bereitzustellen, bedarf es einer geeigneten Logistik und entsprechender Prozesse, die sicherstellen, dass in jedes Fahrzeug die dafür vorgesehene Software in der richtigen Version gelangt. In [FA03], [AL03] und [KK03] wird jeweils die Prozesskette für die Update-Programmierung bei Audi, Volvo Cars und DaimlerChrysler vorgestellt. Alle Beiträge sehen eine durchgängige Prozess- und Logistikkette und das Erfassen, Verwalten und Überprüfen von Abhängigkeiten zwischen Hardware und Software-Modulen als die kritischen Faktoren für einen erfolgreichen flächendeckenden Einsatz der Flash-Programmierung an. In diesem Beitrag wird eine Software-Architektur für den Software-Download vorgestellt, die sich an die unterschiedlichen Gegebenheiten in Entwicklung, Produktion und Außenorganisation auf einfache Weise adaptieren lässt. Damit hat die hier vorgestellte Software-Architektur das Potential, die Prozesse für den Software-Download zu vereinheitlichen und damit zu vereinfachen. Das nächste Kapitel skizziert, wie die heutige Reprogrammierung von Steuergeräten durchgeführt wird, und in Kapitel 3 wird die neue Software-Architektur und ein Prototyp zu deren Evaluierung vorgestellt.

2 Heutige Reprogrammierung von Steuergeräten

Zum Flashen von Steuergeräten wird der sogenannte E-Tester oder Werkstatt-Tester, der an den Diagnosestecker des Fahrzeugs angeschlossen wird, eingesetzt. Zwischen Werkstatt-Tester und dem Steuergerät, das reprogrammiert werden soll, kommt das Protokoll KWP2000 [ISO1] zum Einsatz. Sowohl der E-Tester, als auch das verwendete Protokoll KWP2000 waren ursprünglich ausschließlich zur Diagnose der Fahrzeug-Steuergeräte gedacht. Einige Services des KWP2000-Protokolls werden heute für die Reprogrammierung von Steuergeräten verwendet.

Steuergeräte, die flashbar und über ein Fahrzeugsubnetz erreichbar sind, integrieren einen Flashloader. Der Flashloader interpretiert die KWP2000-Nachrichten, die zur Reprogrammierung an das Steuergerät gesandt werden, und führt die entsprechenden Aktionen wie zum Beispiel das Löschen oder das Beschreiben des Flashspeichers aus. In Bild 2-1 repräsentiert die Download-Steuerung im Steuergerät den Flashloader.

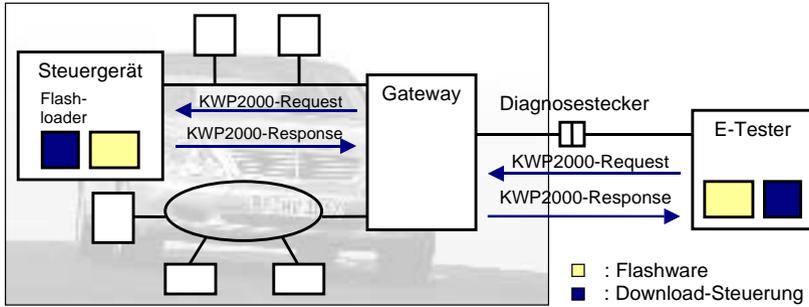


Bild 2-1: Reprogrammierung von Steuergeräten

Der Werkstatt-Tester sendet KWP2000-Request-Kommandos, die über ein oder mehrere Gateways zu demjenigen Steuergerät weitergeleitet werden soll, das reprogrammiert werden soll. Das Steuergerät antwortet zu jedem Request mit einer KWP2000-Response-Nachricht, die angibt, ob der empfangene Request erfolgreich ausgeführt werden konnte.

3 Adaptierbare Software-Architektur und Prototyp

Die Reprogrammierung von Steuergeräten ist heute nur durch den Einsatz des Werkstatt-Testers – einer speziellen Hard- und Software – möglich. Eine größere Flexibilität bei der Reprogrammierung von Steuergeräten kann erreicht werden, wenn die Download-Steuerung des E-Testers (siehe Bild 2-1) durch Software-Komponenten ersetzt wird, die sich je nach Fahrzeugausstattung entweder im Fahrzeug selbst oder in der Infrastruktur befinden können. Befinden sich die Software-Komponenten zur Reprogrammierung im Fahrzeug, so muss lediglich die Flashware (und zugehörige Konfigurationsdaten für den Flashvorgang) über einen beliebigen Übertragungsweg (GSM, GPRS, UMTS, Bluetooth, W-LAN, drahtgebunden) oder ein beliebiges Medium (CD, Memory Stick) in das Fahrzeug transportiert werden. Das Fahrzeug kann dann selbstständig – ohne das Vorhandensein eines Werkstatt-Testers – eine Reprogrammierung der Steuergeräte durchführen. Es ist auch denkbar, dass die Komponenten zur Reprogrammierung bei einem anstehenden Software-Download selbst erst in das Fahrzeug geladen werden. Ist in einem Fahrzeug kein Steuergerät vorhanden, das genügend Speicher und Prozessorleistung besitzt, um die Komponenten für eine Reprogrammierung zu integrieren, so können sich diese in der Infrastruktur beispielsweise in einer Software-Tankstelle, in einem Programmiergerät in der Werkstatt bzw. am Band oder auf einem PC in der Entwicklung befinden.

Bild 3-1 zeigt einen Prototypen für einen telematikmoderierten Software-Download [HS03]. Hier befinden sich die Software-Komponenten zur Reprogrammierung – in der Form von Bundles – auf einem OSGi-Service-Gateway [OS03] in einem Telematiksteuergerät – der Head-Unit. Die Head-Unit ist aus Gründen des vergleichsweise großen Speichers und der verfügbaren Prozessorleistung, sowie der Nähe zu externen Kommunikations-Schnittstellen ein geeigneter Kandidat für die Integration dieser Komponenten. Das OSGi-Service-Gateway ist dabei eine Dienstplattform, die den Lebenszyklus von Komponenten (Bundles) verwalten kann und eine Administration der Dienste von der Infrastruktur aus ermöglicht. In Bild 3-1 stehen die Komponenten Local-Administration und Remote-Administration stellvertretend für diese Verwaltungsfunktionen, die anteilig

auf dem Service-Gateway im Fahrzeug und in der Infrastruktur realisiert sind. Der Einsatz eines OSGi-Service-Gateways im Fahrzeug und eines Management-Server-Frameworks in der Infrastruktur ist nicht zwingend erforderlich, erleichtert aber die Verwaltung von Fahrzeugen und der darauf installierten Software. Das Wireless Communication Module in der Head-Unit und in der Infrastruktur sorgt dafür, dass die Kommunikation zwischen den Komponenten der Head-Unit und der Infrastruktur abgesichert und transparent erfolgt. Dass über eine Luftschnittstelle kommuniziert wird, ist für die Komponenten verborgen.

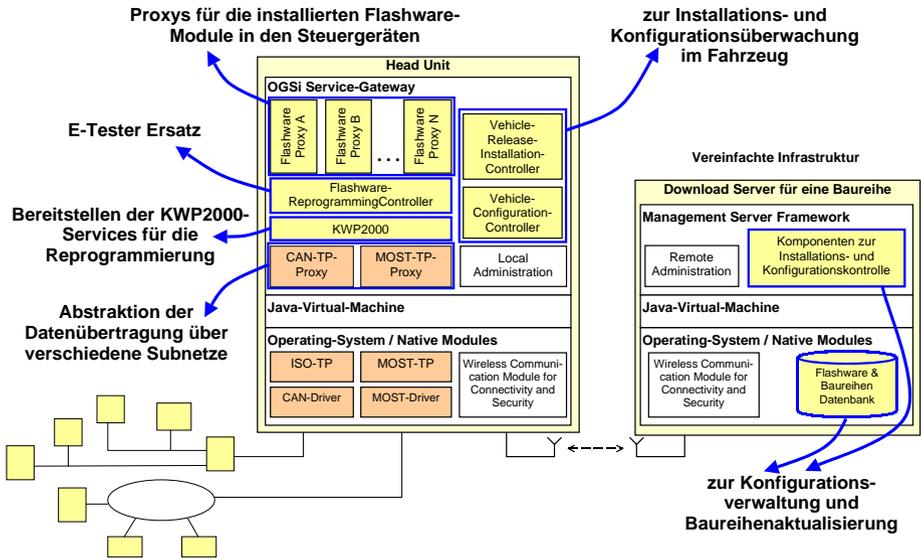


Bild 3-1: Prototyp für einen telematikmoderierten Software-Download

Die essentiellen Komponenten für die Reprogrammierung sind die Transportprotokoll-Proxys (CAN-TP-Proxy, MOST-TP-Proxy), die KWP2000-Komponente und der Flashware-Reprogramming-Controller. Die TP-Proxys bieten dabei der KWP2000-Komponente eine einheitliche Schnittstelle zum Versenden von Nachrichten an und führen die erforderlichen Anpassungen zum Versenden von Transportnachrichten an den jeweiligen Subnetzen durch. Kommt ein neues Fahrzeug-Subnetz (zum Beispiel Flex-Ray oder TT-CAN) zum Einsatz, so muss lediglich ein entsprechender TP-Proxy hinzugefügt werden, um Steuergeräte an diesen Fahrzeug-Subnetzen flashen zu können.

Die Flashware-Proxys (hier in der Form von Bundles) sind Stellvertreter für die auf den Steuergeräten installierten Flashware-Module. Soll ein Flashware-Modul in einem Steuergerät aktualisiert werden, so wird ein neuer Flashware-Proxy, der die neue Flashware und zugehörige Konfigurationsinformation für das Steuergerät enthält, vom Download-Server in der Infrastruktur in das Fahrzeug geladen. Der in das Fahrzeug geladene Flashware-Proxy kontaktiert die Komponenten der Installations- und Konfigurationsüberwachung und veranlasst nach positiver Rückmeldung die Installation der Flashware. Hierzu wird die Flashware und die Konfigurationsinformation vom Flashware-Proxy an den Flashware-Reprogramming-Controller übergeben, der die Konfigurationsdaten interpretiert, die Parameter für die KWP2000-Nachrichten zusammenstellt und diese in der

richtigen Sequenz an das zu programmierende Steuergerät sendet. Die Konfigurationsdaten geben hierbei beispielsweise Aufschluss über den Aufbau der Flashware oder über Spezifikas beim Download-Vorgang in ein spezielles Steuergerät. Nach einer erfolgreichen Installation löscht der Flashware-Proxy die in ihm enthaltene Flashware und die Konfigurationsdaten und wird fortan im Rahmen der Konfigurationsüberwachung im Fahrzeug weiter eingesetzt. Die Komponenten für die Installations- und Konfigurationsüberwachung im Fahrzeug, sowie die Konfigurationsverwaltung in der Infrastruktur werden hier nicht näher betrachtet. Siehe hierzu [HS03].

Muss sowohl ein Flashware-Modul an einem CAN-Subnetz, als auch ein Flashware-Modul in einem MOST-Ring aktualisiert werden, so können diese beiden Aktualisierungsvorgänge parallel durchgeführt werden. Hiermit kann die Zeit für das Flashen gegenüber einer E-Tester-basierten Aktualisierung erheblich reduziert werden. Insbesondere für die Programmierung von Steuergeräten in der Fertigung am Band, kann man sich die Möglichkeit des parallelen Flashens zu Nutze machen. Um am Band möglichst viele Steuergeräte gleichzeitig flashen zu können, kann an allen Fahrzeugsubnetzen (zum Beispiel Motorraum-CAN, Komfort/Karosserie-CAN und MOST-Ring) parallel geflasht werden. Eine weitere Parallelisierung innerhalb eines Fahrzeug-Subnetzes ist möglich, indem so viele Steuergeräte an einem Subnetz gleichzeitig geflasht werden, bis die maximale Buslast erreicht wird. Für die Bandende-Programmierung kann hierzu ein PC zum Einsatz kommen, der den identischen Aufbau wie die Head-Unit in Bild 3-1 hat, und für jedes Fahrzeug-Subnetz, an dem Steuergeräte geflasht werden sollen, einen geeigneten TP-Proxy integriert.

4 Zusammenfassung

Der Ersatz des E-Testers durch Software-Komponenten, die sowohl im Fahrzeug selbst, als auch in der Infrastruktur – beispielsweise bei der Bandende-Programmierung – eingesetzt werden können, bietet nicht nur das Potential, die Prozesskette in der Infrastruktur zu vereinheitlichen und zu vereinfachen, sondern auch die Möglichkeit der parallelen Programmierung zur Reduktion der Flashzeiten.

5 Literaturverzeichnis

- [AL03] H. Alminger, „Downloading software during development, end of line and on the aftermarket, experiences from Volvo Cars“, Software-Flashen im Kfz, Stuttgart, Jan. 2003
- [FA03] M. Faulbacher, „Updateprogrammierung entlang der Prozesskette aus Sicht eines Automobilherstellers“, Software-Flashen im Kfz, Stuttgart, Jan. 2003
- [HS03] C. Heinisch, M. Simons, „Telematikmoderierter Software-Download in Kfz-Steuergeräte“, Entwicklerforum Kfz-Elektronik, Ludwigsburg, Mai 2003
- [ISO1] ISO 14230: „Road vehicles – Diagnostic systems – Keyword Protocol 2000 – Part 3: Application Layer“
- [KK03] T. Kühner, S. Kreuser, „Prozesssicheres Flashen in der Entwicklung, der Produktion und der Außenorganisation bei DaimlerChrysler“, Software-Flashen im Kfz, Stuttgart, Jan. 03
- [OS03] OSGi-Service-Plattform Spezifikation, Release 3, März 2003
- [SC03] H.-E. Schurk, „Spiel ohne Grenzen oder am Band mit unbegrenzten Möglichkeiten? Einführung in die Thematik Flashen“, Software-Flashen im Kfz, Stuttgart, Jan. 2003