



ASCET-SD im Praxis-Einsatz bei Knorr-Bremse

Auch bei Nutzfahrzeugen ermöglicht das Elektronische Stabilitäts-Programm (ESP) in kritischen Fahrsituationen eine wesentliche Verbesserung der Fahrsicherheit. Dabei müssen bei der Software-Entwicklung auch unterschiedliche Ladezustände und Fahrzeugkonfigurationen – etwa mit Anhänger oder Auflieger – berücksichtigt werden. Im Rahmen der Entwicklung eines Anhängersteuermoduls wurde bei Knorr-Bremse mit ASCET-RSF bereits der erste Schritt in Richtung eines neuen Entwicklungsprozesses unternommen. Die Serienentwicklung eines ESP-Systems für Nutzfahrzeuge wurde nun in einem Pilotprojekt mit dem aktuellen Embedded Control Entwicklungssystem ASCET-SD der Firma ETAS durchgeführt.

1 Einleitung

Die Disziplin des Software-Engineering ist im Vergleich zu anderen Ingenieurdisziplinen sehr jung und unterliegt aus diesem Grund noch großen Veränderungen. Die steigende Komplexität der Software steht dabei meist konträr zur Software-Qualität und zu einer möglichst kurzen Entwicklungszeit. Analog zu anderen Entwicklungsbereichen versucht man bei der Software-Entwicklung dieser Diskrepanz durch den Einsatz automatisierender Methoden und Verfahren zu begegnen.

Eine besondere Stellung nimmt dabei die Software-Entwicklung für eingebettete Systeme ein. Im Vergleich zu PC-basierten Systemlösungen besitzen eingebettete –

oder embedded-Systeme in der Regel eine einfache Benutzerschnittstelle und weniger komplexe Programmstrukturen. Erschwerende Randbedingungen für den Entwurf eingebetteter Systeme sind jedoch Anforderungen an das Echtzeitverhalten, die Systemsicherheit und die Robustheit gegenüber externen Einflüssen. Die nicht standardisierte, anwendungsspezifische Hardware behindert zudem eine stärkere Automatisierung der Software-Entwicklung.

So wird bis heute in vielen Anwendungen hardwareabhängiger Assemblercode implementiert. Einerseits ist dies durch die Kenntnisse und Gewohnheiten der Entwickler begründet, andererseits wurde der Assemblereinsatz durch die Restriktionen an die Anwendungssoftware bezüglich

Speicheranforderungen und Laufzeitverhalten erzwungen.

Die offensichtlichen Nachteile der Assemblerprogrammierung – etwa die geringe Portabilität und Wiederverwendbarkeit sowie die schlechte Wartbarkeit (Lesbarkeit) – führte in der Vergangenheit zum verstärkten Einsatz höherer Programmiersprachen wie C. Dieser Trend wurde mit der zunehmenden Verfügbarkeit und Zuverlässigkeit leistungstarker Hochsprachen-Compiler für die unterschiedlichen Zielprozessoren noch verstärkt.

Die zunehmende Software-Komplexität erfordert den Einsatz eines festgelegten organisatorischen Rahmens für den Software-Entwicklungsprozess. Ein solches Prozessmodell stellt das V-Modell dar, das neben

Der Verfasser

Dr. Detlef Zerfowski ist bei der Knorr-Bremse SfN GmbH Beauftragter für die Software-Qualität. Außerdem zählt die Entwicklung einer Plattformsoftware für ESP sowie die Einführung eines Standort-übergreifenden Software-Entwicklungsprozesses zu seinen Aufgabengebieten.



den eigentlichen Entwicklungsschritten auch die Verifikation und Validation der Software beinhaltet, **Bild 1**.

Der Entwicklungsprozess ist dabei in mehrere Phasen unterteilt. Beginnend mit einer Anforderungsdefinition beziehungsweise Spezifikation wird über einen Grob- und Feinentwurf zur Modulimplementierung übergegangen. In jeder Entwurfsphase werden Testfälle definiert, die die einzelnen Entwicklungsschritte überprüfbar halten.

2 Einsatz von ASCET-RSF beim Anhängersteuermodul

Im Rahmen der Entwicklung eines Anhängersteuermoduls (ASM) ging Knorr-Bremse den ersten Schritt in Richtung eines neuen Entwicklungsprozesses. Dabei kam ASCET-RSF zum Einsatz. Bei dem Vorgänger des aktuellen ASCET-SD handelte es sich um ein CASE-Werkzeug, das die graphische Spezifikation der zu implementierenden Regelungsalgorithmen und deren Entwicklung erlaubte. Es bestand jedoch noch keine Möglichkeit der automatischen Codegenerierung und erforderte somit für die Zielhardware eine manuelle Reimplementierung der Funktionen in C.

Neben der zeitaufwändigen Umsetzung in C ergab sich als entscheidender Nachteil, dass das in ASCET-RSF vorliegende Modell, bedingt durch funktionale Änderungen im implementierten C-Code, binnen kürzester Zeit nicht mehr aktuell war, da die Nachführung der Änderungen im ASCET-RSF-Modell nicht zu erzwingen war. Der anfangs vorhandene Vorteil einer guten graphischen Spezifikation der Steuerungssoftware war somit nicht mehr gegeben. Als Fazit ergab sich die Notwendigkeit eines durchgängigen Entwicklungsprozesses von der Spezifikation bis zum Seriencode.

3 Pilotprojekt Elektronisches Stabilitäts-Programm

Aufgrund dieser Problematik entschied sich Knorr-Bremse im Rahmen eines Pilotprojektes – der Serienentwicklung eines aus dem Pkw-Sektor hinlänglich bekannten Elektronischen Stabilitäts-Programms (ESP) im Nutzfahrzeubbereich – für den Einsatz von ASCET-SD.

3.1 ESP für Nutzfahrzeuge

Bei einem ESP für Nutzfahrzeuge müssen im Vergleich zum ESP für Personenwagen zusätzliche Einflussgrößen – wie etwa un-

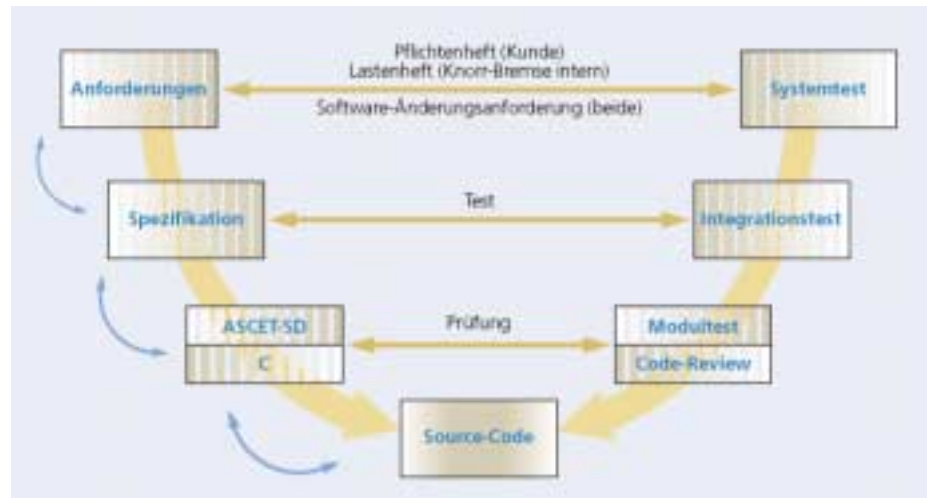


Bild 1: Die zunehmende Software-Komplexität erfordert den Einsatz eines festgelegten organisatorischen Rahmens für den Software-Entwicklungsprozess

terschiedliche Ladezustände und Fahrzeugkonfigurationen mit Anhänger oder Auflieger – berücksichtigt werden. Dabei ist das ESP eine Teilkomponente des Elektronischen-Brems-Systems (EBS), **Bild 2**. Das ESP-Steuergerät kommuniziert über den Bremsen-CAN mit dem EBS-Steuergerät. Über entsprechende Schnittstellen hat das ESP somit Eingriffsmöglichkeiten auf die im EBS-System integrierten Funktionen des Anti-Blockier-Systems (ABS) und der Antriebs-Schlupf-Regelung (ASR).

Weiterhin besitzt das von Knorr-Bremse entwickelte ESP ein Dynamisches-Stabilitäts-Programm (DSP), das auf mittlerem und niedrigem Reibwert ein Einknicken (Jack-Knifing) von Zugmaschine und Auflieger verhindert und das Unter- und Übersteuern von Zugmaschinen sowie Zugmaschinen mit Hängern innerhalb der physikalischen Möglichkeiten unterbindet. Hierzu bremst

das DSP selektiv einzelne Räder ab. So wird im Fall des Untersteuerns hauptsächlich das hintere innere Rad eingebremst. Außerdem ist auch eine Roll-Over-Prevention (ROP) implementiert. Diese vermeidet im Rahmen der physikalischen Grenzen das Umkippen des Fahrzeugs.

Das Vorhandensein mehrerer Regelungsfunktionen kann dazu führen, dass konkurrierende Regeleinriffe angefordert werden. Hieraus ergibt sich die Notwendigkeit entsprechend angepasster Regelstrategien. Da es sich bei dem elektronischen Bremsensystem um eine sicherheitskritische Steuerungskomponente handelt, werden zu den bereits erwähnten Funktionen zusätzliche Sicherheits- und Überwachungsfunktionen hinzugefügt. Diese sollen sowohl hardwarebedingte Fehlfunktionen als auch Fehler auf der Sensor- und Aktuatorebene aufdecken.

Summary

ASCET-SD in Practical Use by Knorr-Bremse

The electronic stability program (ESP) also enables increased driving safety for commercial vehicles in critical driving situations. For this purpose, different load states and vehicle configurations – for example with a trailer or semitrailer – have to be taken into consideration during software development. Knorr-Bremse used ASCET-SD, the ETAS embedded control development system, in a pilot project for the production development of an ESP system for commercial vehicles.

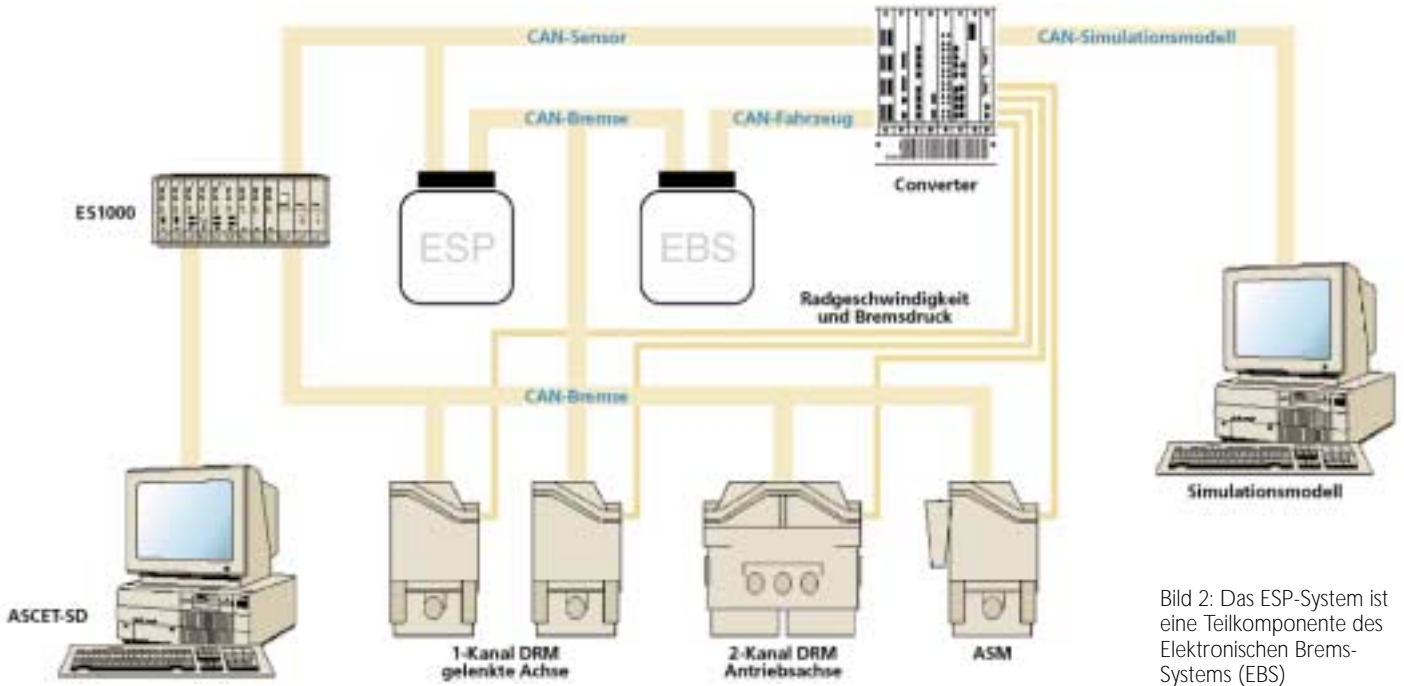


Bild 2: Das ESP-System ist eine Teilkomponente des Elektronischen Brems-Systems (EBS)

3.2 Software-Architektur des ESP

Ausgangspunkt bei der Realisierung des ESP war die Strukturierung der Software-Architektur wie in **Bild 3** dargestellt. Es ergab sich dabei eine natürliche Aufteilung der Software-Komponenten in applikationsnahe Module, die über definierte Schnittstellen Zugriff auf hardwarenah programmierte Module – der sogenannten Software-Plattform – besitzen. Letztere dient der Kapselung der Zielhardware und bietet dünne Schnittstellen für Zugriffe auf die Zielhardware, um ein hohes Maß an Wiederverwendbarkeit der Applikationsfunktionen und eine einfachere und schnellere Übertragung der Software-Plattform auf andere Zielrechner zu gewährleisten.

Es wurde jedoch nicht die gesamte Steuergerätesoftware in ASCET-SD modelliert. Da

die Vorteile dieses Entwicklungswerkzeugs im Bereich der Applikationsfunktionen liegt, wurde die Software-Plattform manuell in C codiert. Von besonderer Bedeutung war dabei eine durchdachte Strukturierung und Modularisierung der Plattform-Software, wodurch eine Sammlung von unabhängigen C-Modulen entstand. Diese sind jeweils einzelnen funktionalen Komponenten – etwa Digital-I/O, A/D-Converter, SSC-Kommunikation, CAN-Kommunikation, EEPROM-Service-Routinen – der verwendeten CPU und der sie umgebenden Hardware zugeordnet.

Die in der Software-Plattform generierten Objekt-Dateien werden in ASCET-SD als externer Objekt-Code eingebunden und die einzelnen Routinen über die in ASCET-SD verwaltete Taskliste aufgerufen. Als Echtzeit-Betriebssystem kommt dabei ERCOSSM der Firma ETAS zum Einsatz.

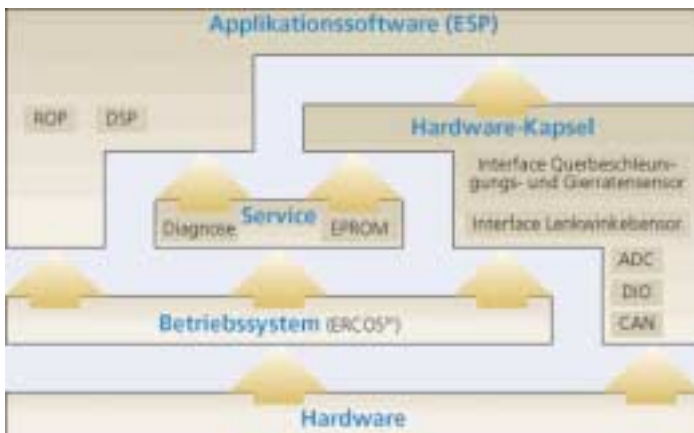


Bild 3: Strukturierung der Software-Architektur

3.3 Funktionsentwicklung unter ASCET-SD

Entscheidend für den Einsatz von ASCET-SD war die erstmals zur Verfügung stehende Durchgängigkeit des Entwicklungsprozesses von der Spezifikation bis zum Seriencode. Somit können bereits in einer frühen Prototypenphase durch Simulation Funktionen auf ihr korrektes Verhalten hin überprüft werden, ohne dass schon eine konkrete Zielhardware vorliegt. Mögliche Designfehler können somit in einem sehr frühen Stadium aufgedeckt und behoben werden.

Dieser sogenannte Fullpass-Betrieb erlaubt Funktionen im Fahrzeug zu testen, ohne dass eine Ziel-Hardware vorhanden ist. Durch die ständige Fortentwicklung der Funktionen innerhalb dieses Entwicklungssystems ist kein Übergang zwischen Entwicklungswerkzeugen mehr notwendig, der zu einem Verlust vorhergehender Implementierungsinformation führen würde.

Die geschilderten Schritte decken im V-Modell die Phasen bis hin zum Feinentwurf ab. Der abschließende Schritt der Implementierung im Steuergerätecode wird durch die automatische Codegenerierung von ASCET-SD realisiert. Die zeitaufwändige Reimplementierung, wie bei der Verwendung von ASCET-RSF, fällt nicht an und verkürzt dadurch die Entwicklungszeit. Die vorliegenden, abstrakteren Implementierungen in ASCET-SD bilden weiterhin die Basis für die Entwicklung und bleiben somit aktuell.

3.4 Prüfstand mit ASCET-SD und ES1000

Eine besondere Bedeutung in der Software-Entwicklung nimmt die Testphase unter realitätsnahen Bedingungen ein. Da Software-Tests in Fahrzeugen zeitaufwändig, kostenintensiv und zudem schlecht reproduzierbar sind, wurde bei Knorr-Bremse ein Prüfstand unter Verwendung von ASCET-SD aufgebaut, **Bild 4**.

Der Prüfstand ermöglicht eine realistische Simulation von Fahrmanövern unter verschiedenen Bedingungen. Außerdem kommt noch ein von Knorr-Bremse entwickeltes Fahrzeugsimulationsmodell auf einem separaten PC zum Einsatz. Dieses Modell erlaubt die Spezifikation unterschiedlicher Fahrzeugkonfigurationen, Umwelteinflüsse und Fahrmanöver, die in Echtzeit anhand eines sich bewegenden Fahrzeuges als 3D-Gittermodell visualisiert werden.

Das Simulationsmodell liefert über den Simulation-Modell-CAN die Daten an den Konverter, die in einem realen Fahrzeug seitens der Sensoren zur Verfügung gestellt würden. Der Konverter realisiert neben der Umsetzung der Simulationsdaten auf den Sensor-CAN und Fahrzeug-CAN auch die Konvertierung der von den im Prüfstand

verbauten Druckregelmodulen (PCM, Pressure Control Module) und Anhängersteuermodul (ASM, engl.: Trailer Control Module, TCM) gelieferten Analogsignale.

Im Prüfstand wird als Hardware-in-the-Loop ein ES1000-System mit ASCET-SD eingesetzt, das selektiv Teilfunktionen der ESP-Software außerhalb des ESP-Steuergeräts in der Experimental-Hardware ablaufen lassen kann (Bypass-Betrieb). Somit ist es möglich, ESP-Funktionalität zum Beispiel in noch nicht-quantisierter Implementierung im Gesamtsystem zu testen. Die dafür notwendige Kommunikation zwischen dem ESP-Steuergerät und der ES1000-Hardware erfolgt dabei über einen Sensor-CAN.

ASCET-SD-Systeme können über eine Hardware-in-the-Loop Einbindung auch direkt im Fahrzeug eingesetzt werden. Die so bei realen Fahrmanövern gewonnenen Messdaten dienen dann anschließend als Grundlage für die am Prüfstand durchzuführenden Testfahrten.

Systeme aus. Dies ist auch mit einem entsprechenden Return-On-Invest (ROI) verbunden.

Die Durchgängigkeit des Entwicklungsprozesses „erzwingt“ einen strukturierten Entwurf sowie eine stets aktuelle Dokumentation der entwickelten Funktionen. Dieses wird durch die graphische Strukturierung der Funktionsblöcke sowie der automatischen Dokumentationsmöglichkeiten erreicht. Die Funktionsentwicklung kann sich weitestgehend auf den Entwurf der reinen Funktionalität konzentrieren und benötigt nicht mehr das detaillierte Wissen über die Zielhardware.

Als Ergebnis wird eine im hohen Maße wiederverwendbare, weil auch getestete Funktionssoftware zur Verfügung stehen. Durch die ebenfalls stark modularisierte, mit schnellen Schnittstellen versehene Software-Plattform wird deren Wiederverwendbarkeit und die schnelle Portierbarkeit auf andere Zielprozessoren ebenfalls gewährleistet. Unter Ausnutzung der bei der ESP-Entwicklung gesammelten Erfahrungen muss sich die Leistungsfähigkeit des gesamten Entwicklungsprozesses zur Zeit im Rahmen der Entwicklung eines neuen EBS-Systems auf einer anderen Zielhardware beweisen.

4 Erwartungen und Aussichten

Durch den Einsatz von ASCET-SD geht Knorr-Bremse von einer deutlichen Effizienz- und Qualitätssteigerung im Bereich der Software-Entwicklung für eingebettete

Bild 4: Prüfstand unter Verwendung von ASCET-SD bei Knorr-Bremse

