

Antrag zur Durchführung einer Projektgruppe

1 Thema

AutoLab – Eine Experimentierplattform für automotive Softwareentwicklung

2 Zeitraum

WS 2007/2008 und SS 2008

3 Umfang

Jeweils 8 SWS

4 Veranstalter

Olaf Spinczyk	ab 1.9. Professor, LS12
Jochen Streicher	ab 1.10. Wiss. Mitarbeiter, LS12
Horst Schirmeier	ab 1.9. Wiss. Mitarbeiter, LS12

Bei Rückfragen vor dem 1.9. zögern Sie bitte nicht, uns unter **os@cs.fau.de** bzw. **09131-8527906** anzusprechen.

5 Aufgabe

5.1 Einleitung

Moderne Kraftfahrzeuge sind heterogene, verteilte, eingebettete Rechnersysteme auf Rädern. Der in Abbildung 1 gezeigte Maybach verfügt beispielsweise über 76 Steuergeräte (engl. *Electronic Control Units* – ECUs) auf Basis von 8-32 Bit Mikrocontrollern, die über nicht weniger als fünf Bussysteme miteinander vernetzt sind. Man geht heute davon aus, dass etwa 40% der Produktionskosten eines Autos in die Fahrzeugelektronik und die auf den Steuergeräten laufende Software fließen [5].

Der Grund für diese enorm großen und stetig wachsenden Aufwendungen liegt in der immer größer werdenden Bedeutung moderner Fahrzeugfunktionen, die ohne Einsatz von Software technisch nicht mehr beherrschbar wären. Typische Beispiele sind über mehrere Steuergeräte verteilt realisierte Fahrzeugfunktionen wie die elektronische Parkbremse oder der automatische Abstandsregeltempomat. Man schätzt, dass Fahrzeuge der Premium-Klasse heute mit Software im Umfang von etwa 10.000.000 Programmzeilen ausgestattet sind. Seit dem Einzug von Software ins Automobil vor ca. 30 Jahren stieg diese Zahl exponentiell, und der Trend wird wohl auch in den nächsten Jahren noch anhalten.

Obwohl in der Summe etliche Megabytes an Software im Automobil bewegt werden, dürfen keine Ressourcen verschwendet werden. Um dem hohen Konkurrenz- und damit Kostendruck in der Automobilbranche Rechnung zu tragen, wird daher dedizierte Hardware eingesetzt und die Systemsoftware anwendungsspezifisch konfiguriert. Zudem werden typischerweise maschinennahe Programmiersprachen wie C oder Assembler eingesetzt.

Die kontinuierlich steigende Komplexität der Software im Kraftfahrzeug und die oben genannten zusätzlichen Erschwernisse, die für heterogene, verteilte, eingebettete Systeme typisch sind, stellen die Automobilhersteller und ihre Zulieferer vor große Herausforderungen. Ohne eine Neuorientierung bezüglich der eingesetzten Entwicklungsprozesse, Entwicklungswerkzeuge, Programmiersprachen und Testverfahren wird es wohl kaum möglich sein, die derzeit stetig wachsende Zahl der Pannen, die durch Elektronik und insbesondere Softwarefehler verursacht werden, in den Griff zu bekommen [6].

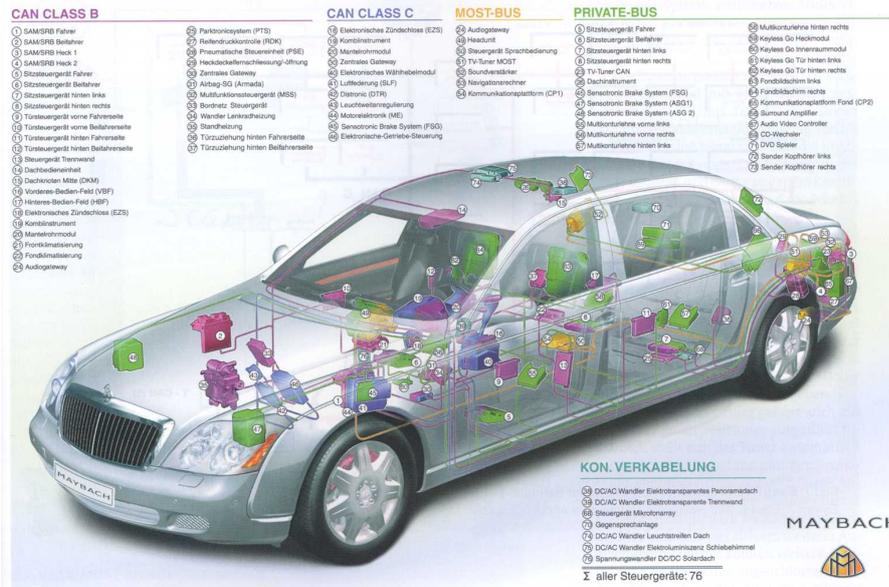


Abbildung 1: Vernetzte Steuergeräte in einem modernen Kraftfahrzeug [4]

5.2 Wissenschaftlicher Kontext

Inzwischen hat sich “Automotive Software Engineering” als eigenständiges Forschungsgebiet herausgebildet, welches bereits einige Ansätze zur Bewältigung dieser Probleme hervorgebracht hat [13]. Zur Beherrschung der steigenden Komplexität werden neben etablierten Methoden der Dekomposition und Trennung der Belange auch neuere, modellbasierte Entwicklungsansätze untersucht [5]. Der riesigen Variantenvielfalt automotiver Software und deren Konfigurierbarkeit versucht man durch systematische Wiederverwendung mit Hilfe von Software-Produktlinien [11] Herr zu werden.

Die neue Arbeitsgruppe “Entwicklung und Betrieb eingebetteter und vernetzter Systeme” ist in diesem Zusammenhang speziell im Bereich der automotiven Systemsoftware aktiv (z.B. [16]). Ziel ist es, neue Entwicklungsmethoden zu erarbeiten und zu bewerten, die es erlauben, hochgradig anwendungsspezifisch konfigurierbare Systemsoftware zu bauen, die trotz Wiederverwendbarkeit und Wartbarkeit die extrem harten Effizianzorderungen erfüllt. Zum einen setzen wir dabei auf Merkmalsbasierte Produktleitung [15], einer Technik, die die systematische Entwicklung von Software-Produktlinien unterstützt. Ein weiterer wichtiger Ansatz ist die Aspektorientierte Programmierung [8] (mittels AspectC++ [14]), ein Entwurfs- und Implementierungsansatz, der eine bessere Modularisierung des Programmcodes durch Trennung der Belange erlaubt und so die Konfigurierung der Software erleichtert.

5.3 Ziele und beabsichtigte Lerneffekte

Um die eigenen Systementwicklungen besser bewerten zu können und generelle Probleme automotiver Systemsoftware zu untersuchen, soll innerhalb der nächsten Jahre ein Labor entstehen, für das die Projektgruppe AutoLab den Grundstein legen soll. Ziel ist dabei, einen realistischen “Brettaufbau” einer Fahrzeugelektronik vorzunehmen und diesen in eine Steuerungs- und Messumgebung einzubetten. Damit könnten dann folgende Einzelziele erreicht werden:

- Schaffung einer Entwicklungsplattform, die Analysen unter realistischen Lastverhältnissen erlaubt.
- Betrachtung des Fahrzeugs als Gesamtheit. Globale Gesichtspunkte wie Lastverteilung (u.a. bei der Flash-Programmierung), Energieverbrauch und Startzeiten könnten untersucht werden.
- Untersuchung alternativer Fahrzeugtopologien.

Um sicherzustellen, dass die Experimentierplattform aktuellen und insbesondere auch zukünftigen Fahrzeugstrukturen nahe kommt, wird eine Zusammenarbeit mit einem Industriepartner angestrebt. Erste Gespräche sind bereits erfolgt.

Zunächst sollen sich die Arbeiten auf der Fahrzeugseite auf den Karosseriebereich konzentrieren. Ein CAN-Bus-Netzwerk [7], wie es in diesem Bereich üblich ist, kann mit vergleichsweise wenig Aufwand beschafft und in Betrieb genommen werden. Der Antriebs- und Fahrwerksbereich (heute in der Regel FlexRay-Netze) sowie "Infotainment" (typischerweise MOST-Netze) sollen im Rahmen der Projektgruppe noch nicht betrachtet werden [2, 3].

Stattdessen liegt der Schwerpunkt der Arbeiten auf der Steuerungs- und Messumgebung. Mit Hilfe dieser Umgebung soll das komplette Fahrzeug zügig programmiert, gestoppt und gestartet sowie beobachtet werden können. Bei einer Beobachtung sollen Ausgabesignale der Steuergeräte und möglichst auch deren Energieverbrauch sowie Bus-Nachrichten zeitgenau erfasst werden. Zudem sollen Eingabesignalleitungen der Steuergeräte programmiert und Testabläufe zum Zwecke der Automatisierung aufgezeichnet werden können. Für die technische Realisierung der Umgebung ist ein zeitgesteuertes FlexRay-Netzwerk ange-dacht, dessen Knoten direkt mit den Steuergeräten bzw. der Messhardware verbunden sind. Dazu müssen adäquate Techniken zu Entwurf und Implementierung zeitgesteuerter Echtzeitsysteme eingesetzt werden [9, 10]. Zur Interaktion mit diesem System soll im ersten Ansatz eine konsolenbasierte PC-Anwendung und später eine graphische Benutzerschnittstelle dienen.

Studentinnen und Studenten, die in der Projektgruppe mitarbeiten, bietet sich neben der Erfahrung, eine komplexe Aufgabe im Team zu lösen, die Möglichkeit, die Welt der automotiven Softwareentwicklung kennenzulernen. Dazu gehört Wissen über typische Hard- und Softwarearchitekturen, wichtige Standards wie OSEK [12] und AUTOSAR [1] und generelles Know-How über die Zusammenhänge in dieser gerade in Deutschland sehr wichtigen und erfolgreichen Branche. Auf der praktischen Seite bietet die Projektgruppe Teilnehmern die Gelegenheit, ein verteiltes, heterogenes, eingebettetes System aus Hard- und Software zu konzipieren, auf Beschaffungen Einfluss zu nehmen und letztlich die Software dafür zu entwerfen und implementieren.

5.4 Rechtliche Hinweise

Die Ergebnisse der Projektarbeit inkl. der dabei erstellten Software sollen dem Fachbereich Informatik uneingeschränkt zur freien Forschung zur Verfügung stehen. Darüber hinaus sind keine Einschränkungen der Verwertungsrechte an den Ergebnissen der Projektgruppe vorgesehen. Es ist jedoch möglich, dass im Rahmen von Industriekooperationen im Einzelfall (z.B. bei Zugriff auf Quellcodes kommerzieller Embedded-Betriebssysteme) Vertraulichkeitsvereinbarungen unterzeichnet werden müssen.

6 Teilnahmevoraussetzungen

Notwendige Voraussetzungen

- Vorlesung "Rechnerarchitektur", "Prozessrechnerarchitektur" oder "Eingebettete Systeme"
- Bereitschaft zur hardware- und systemnahen Programmierung sowie keine Angst vor Hardware
- Verständnis englischsprachiger Artikel und Handbücher

Wünschenswert

- Programmierkenntnisse in C/C++

7 Minimalziele

1. Konzept: Detaillierter Entwurf

- (a) eines einfachen Fahrzeugelektronik-Versuchsaufbaus (CAN-basiertes Karosserienetzwerk) und
- (b) einer Steuerungs- und Messumgebung (Starten/Stoppen/Flash-Programmierung; zeitgenau-eres Erfassen von Änderungen an digitalen Ausgabeleitungen und CAN-Bus-Nachrichten) mit konsolenbasierter Benutzerschnittstelle

2. Inbetriebnahme des Versuchsaufbaus: Vernetzung der Steuergeräte und Inbetriebnahme mit Hilfe der dazugehörigen Entwicklungswerkzeuge und Standardsoftware

3. Steuerungs- und Messumgebung: Implementierung der entworfenen Umgebung in Hard- und Software
4. Demonstrator: Entwurf und Implementierung einer geeigneten Beispielanwendung für das Fahrzeugnetz, die in der Lage ist, die Funktionstüchtigkeit des Fahrzeug-Versuchsaufbaus und die zuvor spezifizierten Eigenschaften der Steuerungs- und Messumgebung zu demonstrieren
5. Dokumentation aller Ergebnisse, insbesondere in Form des Zwischen- und Abschlussberichts

8 Literatur

- [1] AUTOSAR homepage. <http://www.autosar.org/>.
- [2] FlexRay Consortium homepage. <http://www.flexray.com>.
- [3] MOST Cooperation homepage. <http://www.mostnet.de>.
- [4] Daimler Chrysler AG. Der neue Maybach. *ATZ/MTZ Sonderheft*, page 125, 2002.
- [5] Manfred Broy. Challenges in automotive software engineering. In Leon J. Osterweil, H. Dieter Rombach, and Mary Lou Soffa, editors, *ICSE*, pages 33–42. ACM, 2006.
- [6] Ferdinand Dudenhöffer. Studie: Ausfallsicherheit der Fahrzeugelektronik. *Automotive*, 03-04/2004:12–15, 2004.
- [7] Konrad Etschberger. *Controller Area Network*. Carl Hanser Verlag, 2002.
- [8] G. Kiczales, J. Lamping, A. Mendhekar, C. Maeda, C. Lopes, J.-M. Loingtier, and J. Irwin. Aspect-oriented programming. In M. Aksit and S. Matsuoka, editors, *Proceedings of the 11th European Conference on Object-Oriented Programming (ECOOP '97)*, volume 1241 of *Lecture Notes in Computer Science*, pages 220–242. Springer-Verlag, June 1997.
- [9] Hermann Kopetz. *Real-Time Systems: Design Principles for Distributed Embedded Applications*. Kluwer Academic Publishers, 1997.
- [10] Peter Marwedel. *Eingebettete Systeme*. Springer-Verlag, 2007.
- [11] Linda M. Northrop and Paul C. Clements. An introduction to software product lines. In Robert L. Nord, editor, *SPLC*, volume 3154 of *Lecture Notes in Computer Science*, page 322. Springer, 2004.
- [12] OSEK/VDX Group. *Operating System Specification 2.2.3*. OSEK/VDX Group, February 2005. <http://www.osek-vdx.org/>.
- [13] Martin Rappl, Alexander Pretschner, Christian Salzmann, and Thomas Stauner. 3rd international workshop on software engineering for automotive systems - seas 2006. In *ICSE '06: Proceeding of the 28th international conference on Software engineering*, pages 1034–1034, New York, NY, USA, 2006. ACM Press.
- [14] Olaf Spinczyk, Daniel Lohmann, and Matthias Urban. Advances in AOP with AspectC++. In Hamido Fujita and Mohamed Mejri, editors, *New Trends in Software Methodologies, Tools and Techniques (SoMeT '05)*, number 129 in *Frontiers in Artificial Intelligence and Applications*, pages 33–53, Tokyo, Japan, September 2005. IOS Press.
- [15] Olaf Spinczyk and Holger Papajewski. Using feature models for product derivation. In *Proceedings of the 10th Software Product Line Conference (SPLC '06)*, Baltimore, Maryland, August 2006.
- [16] Olaf Spinczyk, Wolfgang Schröder-Preikschat, Danilo Beuche, and Holger Papajewski. PURE/OSEK - Eine aspektorientierte Betriebssystemfamilie für Kraftfahrzeuge. In *Proceedings of the GI Workshop on Automotive SW Engineering & Concepts*, pages 330–334, Frankfurt/Main, Germany, September 2003. in *INFORMATIK 2003 (GI Lecture Note in Informatics)*, ISBN 3-88579-363-6.