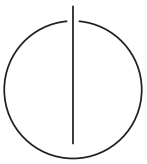
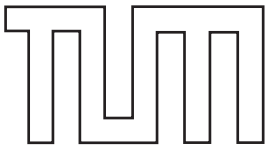


TECHNISCHE UNIVERSITÄT MÜNCHEN
FAKULTÄT FÜR INFORMATIK

Software & Systems Engineering
Prof. Dr. Dr. h.c. Manfred Broy



SPES 2020 Deliverable D1.4.C-3b

SPES Beispiel Automotive: Motorsteuerung (Durchstichprototyp in AutoFocus3)

Author: Florian Hölzl

Version: 1.0
Date: December 2, 2010
Status: Released

Technische Universität München - Fakultät für Informatik - Boltzmannstr. 3 - 85748 Garching

Contents

1 Einführung **3**
- 1.1 Mechanik und Regelungsaufgabe . . . 3
- 1.2 Messeinrichtung und Stellglied . . . 3
- 1.3 Sollwertbestimmung . . . 4
- 1.4 AutoFocus 3 . . . 4
- 1.5 Architekturperspektiven . . . 5

2 Funktionale Architektur **5**

3 Logische Architektur **6**
- 3.1 High-level Architektur Gesamtsystem . . . 6
- 3.2 Detaillierte Architektur Motorsteuerung . . . 7

4 Technische Architektur **9**

5 Allokation von Funktionen, Deployment von Komponenten, Verbau von ECUs **9**
- 5.1 Allokation von Funktionen . . . 10
- 5.2 Deployment von Komponenten . . . 10
- 5.3 Verbau von ECUs . . . 11

6 Abschließende Bemerkungen **11**

References **12**

1 Einführung

Dieses Dokument beschreibt den Durchstichprototypen einer elektronischen, software-basierten Motorsteuerung, der mit den SPES Systemperspektiven und dem Modellierungswerkzeug AutoFocus 3 konzipiert wurde.

Wir betrachten die Perspektiven *Funktionsarchitektur*, *Logische Architektur* und *Technische Architektur* sowie *Allokation* und *Deployment* zwischen diesen. Dabei liegt unser Fokus auf den aus Informatiksicht relevanten Artefakten, weniger auf den regelungstechnischen Details. Als Basis dient das von Robert Bosch GmbH zur Verfügung gestellte Anforderungsdokument [Hei10] sowie dem Buch "Dieselmotor-Management" [Bau98] aus der gelben Reihe von Bosch.

Im folgenden beschreiben wir kurz die Regelungsaufgabe und die mechanischen Komponenten sowie die für die Sollwertbestimmung relevanten Informationen.

1.1 Mechanik und Regelungsaufgabe

Wir verwenden die Bosch Standard-Reiheneinspritzpumpe mit elektronischer Regelung (PE-EDC, [Bau98], S. 204) mit einem Direkteinspritz-Dieselmotor (DI). Als Reglungsaufgabe stellen sich folgende Teilregelkreise:

- Lageregelung der Pumpenregelstange
- Einspritzmengenregelung gemäß Fahrerwunsch, aktueller Pumpendrehzahl und Drehmomentbedarf der Klimaanlage.

1.2 Messeinrichtung und Stellglied

Wir gehen für den Durchstich von folgenden Messeinrichtungen und Sensoren aus:

- Drehzahlsensor zur Messung der Einspritzpumpendrehzahl.
- Regelwegsensor zur Überprüfung der Regelstange für die Fördermengenregulierung.
- Temperatursensor am Motor zur Erkennung der Kalt-/Warmstartbedingung
- Fahrpedalsensor zur Ermittlung des Fahrerwunsches
- Energiebedarfssignal des Klimageräts

Die Aktuatorik beschränkt sich auf ein einzelnes Stellglied:

- Stellmagnet der Regelstange der Reiheneinspritzpumpe

Die Sensorik und Aktuatorik werden sich in der technischen Architektur (in der Beschreibung des Steuergeräts) wiederfinden. Die jeweiligen Signale finden sich als Ports in der Logischen Architektur. Für das Gesamtsystem werden die logischen Ports auf Sensoren und Aktoren abgebildet (vgl.5).

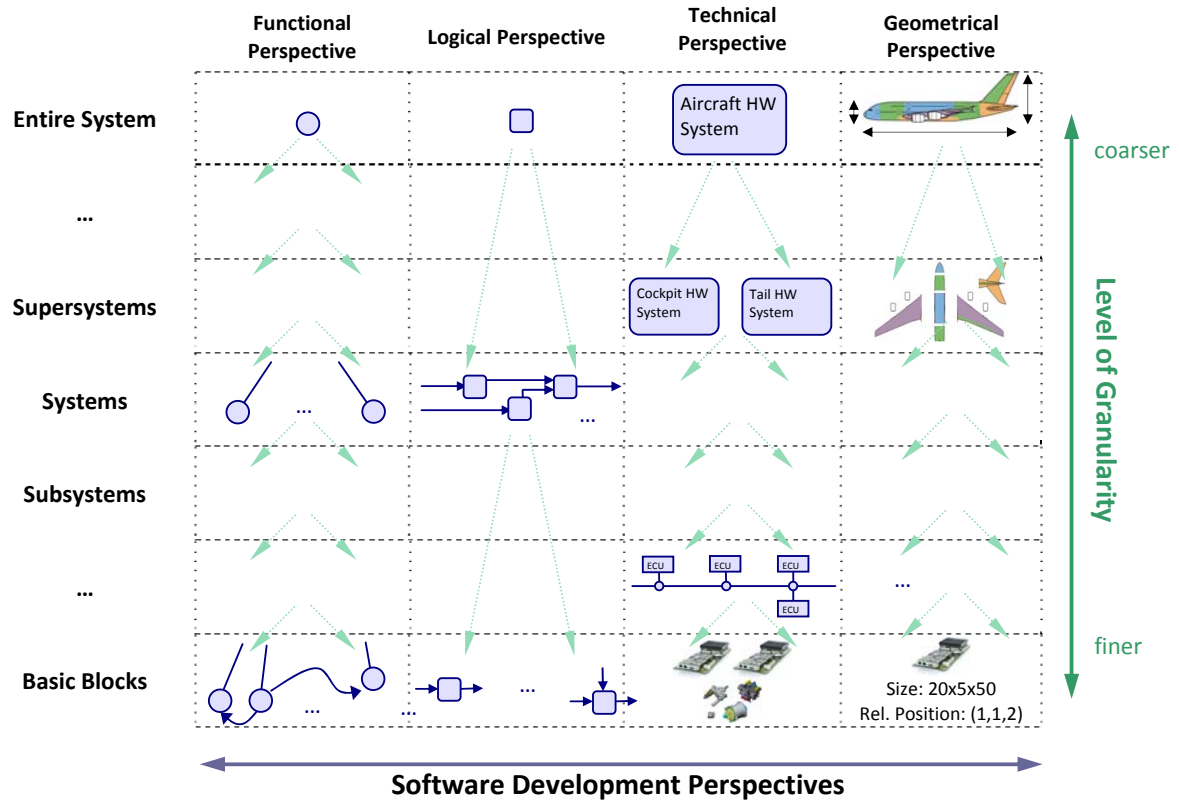


Figure 1: Perspektiven und Granularitätsebenen nach SPES ZP-AP-1

1.3 Sollwertbestimmung

Die Sollwertbestimmung ist abhängig von einer ganzen Reihe von Faktoren, wobei wir nur ein sehr reduziertes System betrachten: Betriebsmodus des Motors, Geschwindigkeitswunsch des Fahrers und Leistungsbedarf des Nebensystems Klimaanlage. Der Durchstichprototyp enthält kein ausformuliertes bzw. ausmodelliertes konkretes Reglerverhalten. Die Implementierung einzelner Reglerverläufe, z.B. für den Vollastbetrieb mit Angleichung sind nicht Gegenstand dieses Dokuments.

1.4 AutoFocus 3

AutoFocus 3 ist ein am Lehrstuhl von Prof. Broy entwickeltes Werkzeug zur Modellierung der Software verteilter, eingebetteter, reaktiver Systeme. Es unterstützt die von SPES ZP-AP-1 entwickelten Modellierungsperspektiven der Funktionalen, Logischen, und Technischen Architektur. Das Werkzeug kann für die gängigen Betriebssysteme unter http://af3.in.tum.de/ frei heruntergeladen werden. Eine Kurzeinführung gibt [HF11]. Eine anwendernahe Einführung in das Werkzeug bietet das AF3 Bilderbuch [HHLP10]. Die mathematische Basis findet sich in [BS01].

1.5 Architekturperspektiven

Eines der wesentlichen Ergebnisse des SPES ZP-AP-1 ist die Matrix der Entwicklungsperspektiven und deren innerer Beziehung. Abb. 1 zeigt die vier Perspektiven in der Horizontalen, sowie die Granularitätsebenen in der Vertikalen.

Im Falle der Motorsteuerung beschränken wir uns auf die ersten drei Perspektiven und lassen die Geometrische Architektur weg. Diese dient im Wesentlichen dazu die Lage von Sensorik und Aktuatorik, sowie der Steuergeräte, räumlich zu beschreiben.

Im Folgenden gehen wir die Modellierung der Motorsteuerung anhand der Perspektiven Funktional, Logisch, Technisch im Einzelnen durch.

2 Funktionale Architektur

Die Funktionale Architektur (FA) modelliert und spezifiziert die wesentlichen Funktionen des zu entwickelnden Systems.

Zum aktuellen Forschungsstand von SPES ZP-AP-1 ist die FA noch nicht in voller Tiefe in AutoFocus 3 integriert. Im Augenblick stellt sie lediglich eine baumartige Funktionsdekomposition (gelbe Kästen) mit der Möglichkeit Abhängigkeiten (dunkle Rauten) zwischen den Funktionen zu spezifizieren. Hierbei werden die wesentlichen Informationen von Funktionen und Abhängigkeiten in textueller Form hinterlegt. Die Interpretation der Baumhierarchie entspricht einer Subfunktionalitätsbeziehung, während die Querbeziehungen sequentielle Aktivierung, alternative Betriebsfunktionen und alternative Softwareanteile (Varianten) erfassen.

Eine Blattfunktion im FA Baum modelliert einen Teil der Gesamtsoftware. In unserem Fall entspricht jede Blattfunktion einem bestimmten Reglerverhalten. Im Werkzeug noch nicht integriert ist die Möglichkeit Funktionen mit einer syntaktischen Schnittstelle und einer Verhaltensspezifikation auszustatten (vgl. SPES Meta-Model [HHK10]).

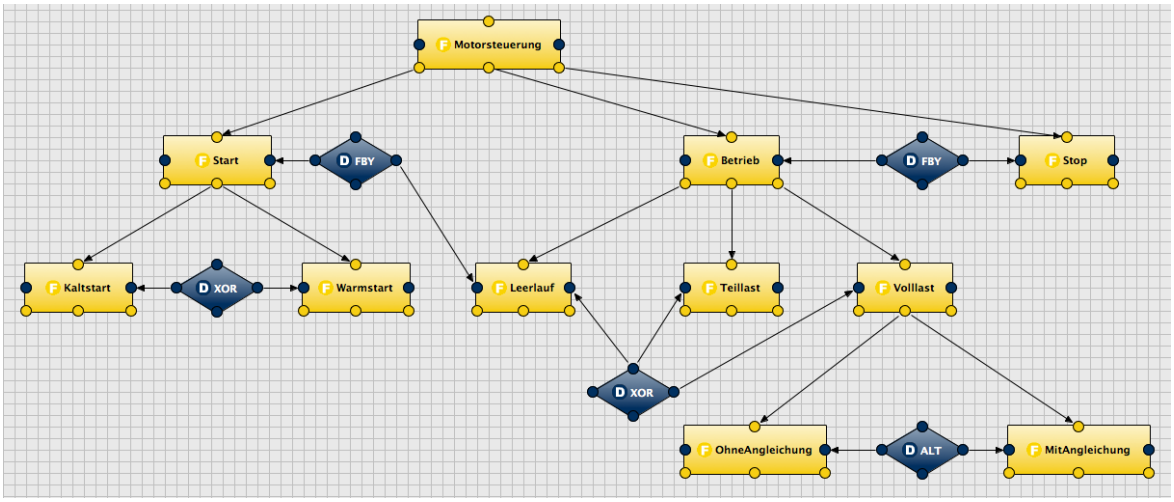


Figure 2: Funktionsarchitektur der Motorsteuerung

Abb. 2 zeigt die Funktionale Architektur der Motorsteuerung. Die Gesamtfunktion der Motorsteuerung unterteilt sich zunächst in die verschiedenen Betriebsphasen des Motors: Start, Fahrbetrieb und Stop. Hierbei definieren die beiden *FBY* (followed-by) sequentielle Abhängigkeiten. Die Motorsteuerung wechselt nach dem Startvorgang in den Leerlaufbetrieb. Bei Beendigung der Betriebsfunktion wechselt die Motorsteuerung zwecks Abschaltung zur Stop Funktion.

Für die Startphase gibt es zwei alternative Funktionen: den Kaltstart und den Warmstart. Die *XOR* Abhängigkeit modelliert hier die alternative Aktivierung (exklusives Oder) der beiden Teilfunktionen.

Die Betriebsfunktion unterteilt sich in drei disjunkte (XOR) Regelungsfunktionen, welche sich durch das vom Motor geforderte Drehmoment unterscheiden: Leerlauf, Teil- und Volllastbetrieb. Für den Volllastbetrieb ergeben sich zwei alternative Regler (ALT Abhängigkeit): mit und ohne Angleichen der Einspritzmenge bei steigender Drehzahl (vgl. [Bau98], S.100f.). Hierbei unterscheiden sich die XOR und die ALT Abhängigkeit. XOR modelliert Funktionen, die sich zur Laufzeit ausschließen, während ALT sich bereits zur Compilezeit ausschließende Funktionen modelliert (Varianten).

3 Logische Architektur

Die logische Architektur (LA) beschreibt die hardwareunabhängige Softwareapplikation in Form von kommunizierenden Komponenten. Hierbei wird ein globaler, synchroner Takt zugrunde gelegt.

Stark kausale Komponenten (blau dargestellt) liefern hierbei eine Antwort jeweils im nächsten Takt nach Empfang einer Eingabe, während schwach kausale Komponenten (gelb dargestellt) unverzögert reagieren. Stark kausale Komponenten haben den Vorteil später beliebig auf Recheneinheiten verteilt zu werden, da die Verzögerung für eine saubere Ausführungssemantik sorgt. Der Preis ist hierfür jedoch, dass sich die Verzögerungen in einer Kette von stark kausalen Komponenten aufaddieren. Schwach kausale Komponenten vermeiden diese addierte Verzögerung, können jedoch nicht mehr beliebig verteilt werden.

Rot dargestellte Komponenten sind zusammengesetzt aus einem Netzwerk von weiteren Komponenten. Hierbei können sich die Verzögerungen je Ausgabeport unterscheiden, je nachdem welche Kausalität die Komponenten des Netzes haben.

3.1 High-level Architektur Gesamtsystem

Die oberste Architekturebene beschreibt, wie in Fig. 3 zu sehen, die wesentlichen Komponenten die Einfluss auf die Motorsteuerung haben. Dies sind zum einen der Fahrer, der durch die Pedalstellung seinen Geschwindigkeitswunsch an die Steuerung übermittelt. Die Einspritzpumpe liefert die Informationen über die aktuelle Stellung der Regelstange und die aktuelle Pumpendrehzahl. Gleichzeitig erhält sie von der Motorsteuerung das Stellsignal für den Regelstangenstellmagnet. Die Klimaanlage liefert der Motorsteuerung einen aktuellen Bedarfswert, der bei der Sollwertbestimmung für die Einspritzmenge berücksichtigt wird. Der Motorblock selbst liefert die Daten eines Temperatursensors zum Zwecke der Bestimmung des Kalt- und Warmstartzustandes. Die eigentliche Motorsteuerung ist in einer Komponente

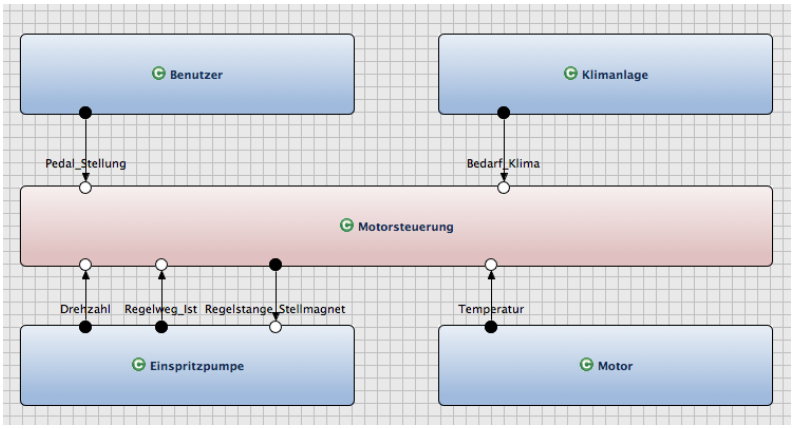


Figure 3: Gesamtarchitektur mit Umgebungskomponenten

gebündelt, die sich jedoch aus einer ganzen Reihe von Unterkomponenten zusammensetzt, die im Folgenden beschrieben werden.

3.2 Detaillierte Architektur Motorsteuerung

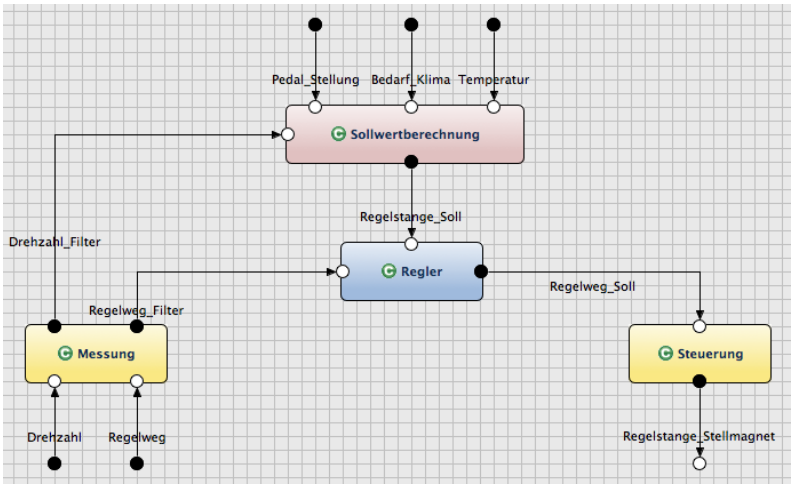


Figure 4: Komponentenstruktur der Motorsteuerung

Fig. 4 zeigt die erste Dekompositionsstufe der Komponente *Motorsteuerung*. Eine Komponente *Messung* erzeugt gefilterte und plausibilisierte Werte der Sensordaten und gibt diese an die Komponente *Sollwertberechnung* und den *Regler* weiter. Die Sollwertberechnung erhält zusätzlich die Signale des Fahrerwunsches, der Motortemparatur und der Klimaanlagenanforderung. Der berechnete Sollwert wird an den Regler weitergegeben und von diesem zusammen mit dem Istwert in ein Stellsignal umgesetzt. Die Komponente *Steuerung* setzt das Stellsignal in eine entsprechende Aktuatoransteuerung um.

Die Signalverzögerung aufgrund der Kausalität beträgt hier für alle Ein-/Ausgabekombinationen einen Takt, da nur die Komponente *Regler* stark kausal ist.

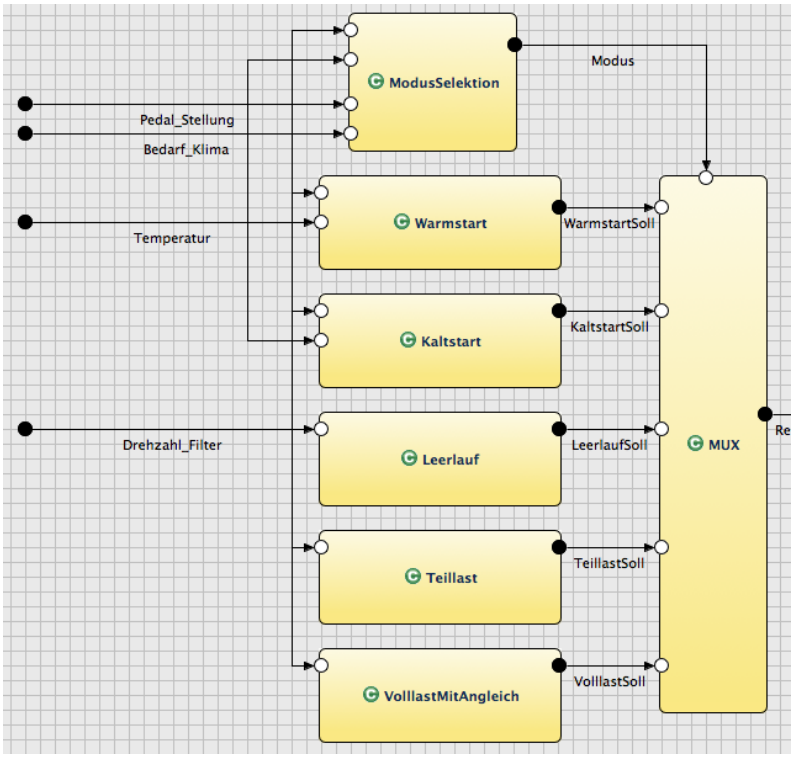


Figure 5: Komponentenstruktur der Sollwertberechnung

Fig. 5 zeigt die Modusarchitektur der Komponente *Sollwertbestimmung*. Hierbei entscheidet die Komponente *ModusSelektion*, welche Fahrsituation vorliegt und steuert die Komponente *MUX* entsprechend an. Dieser Multiplexer schaltet daraufhin den entsprechenden Regler auf den Ausgabeport durch. Die fünf Komponenten unterhalb der Modusselektion implementieren, das zur jeweiligen Fahrsituation gehörende Reglerverhalten, wie es in der Funktionsarchitektur skizziert wurde. Am Beispiel der modusgesteuerten Motorsteuerung zeigt sich hierbei eine 1:1-Abbildung zwischen den Funktionen der FA und den Komponenten der Sollwertbestimmung.

Sämtliche Komponenten der Sollwertbestimmung sind schwach kausal, sodass die Sollwertbestimmung selbst keine logischen Zeitschritt benötigt. Schwache Kausalität erlaubt es Komponenten parallel zu berechnen vorrausgesetzt, dass hierbei die topologische Sortierung beachtet wird. Die Komponenten in der Mitte von Abb. 5 können also parallelisiert werden, während die Multiplexerkomponente *MUX* erst nach dem Rechnen aller anderen beginnen kann.

Hier zeigt sich ein Nachteil der strikt parallelen Komposition von AutoFocus 3. Es werden in jedem logischen Takt alle fünf Sollwertregler berechnet und vier der Ergebnisse durch den Multiplexer anschließend verworfen. Die Rechenlast ist hier also nur unzureichend effizient genutzt. Eine Möglichkeit diesen Nachteil auszugleichen bieten Modusautomaten wie sie in [BBR+05] verwendet wurden. Die Portierung dieser Modellierungstechnik ist Gegenstand aktueller Arbeiten an AutoFocus 3.

4 Technische Architektur

Die Technische Architektur (TA) modelliert die Rechen- und Übertragungseinheiten des Systems. Wir unterscheiden hier elektronische Recheneinheiten (ECUs) von Bussystemen und Netzwerken. Die Verbindung dieser beiden Einheiten werden über Bus- bzw. Netzwerkcontroller hergestellt. Weiterhin können ECUs mit spezifischen Sensoren und Aktuatoren ausgestattet sein.

Die TA ist hierbei durch spezifische Elemente erweiterbar und konkret auszugestalten. Ein Beispiel mit der Ausgestaltung für die Hardware im Automotivelab der TU München findet sich im Bilderbuch [HHLP10] und im Metamodelldokument [HHK10].

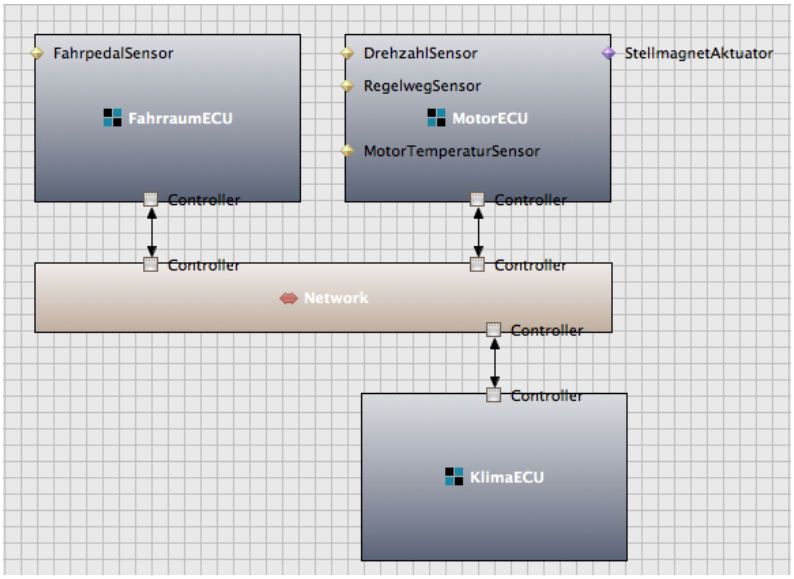


Figure 6: Technische Architektur

Fig. 6 zeigt die TA dert Motorsteuerung. Der betrachtete Ausschnitt des Bordnetzes umfasst eine Recheneinheit im Fahrraum, eine an der Klimaanlage und das Motorsteuergerät. Das Fahrraumsteuergerät hat die Aufgabe die Fahrpedalstellung zu ermitteln und an die Motorsteuerung zu übermitteln. Das Klimasteuergerät übermittelt seine Anforderung über das Netzwerk an die Motorsteuerung. Das Motorsteuergerät selbst liest über direkt angeschlossene Sensorik die Drehzahl, die Regelstangenstellung und die Motortemperatur aus. Über den direkt angeschlossenen Stellmagneten kann das Motorsteuergerät die Regelstange und damit die Einspritzmenge steuern.

5 Allokation von Funktionen, Deployment von Komponenten, Verbau von ECUs

Die bisherigen Abschnitte demonstrierten die jeweilige Perspektive der SPES Matrix. Sowohl bei der FA als auch bei der LA haben wir die vertikale Dekomposition demonstriert. Im Falle der TA ist die Dekomposition nicht-rekursiv und nur eine Ebene flach. Es ist jedoch auch

hier denkbar mehrere Dekompositionsebenen einzuführen. Beispielsweise könnte eine ECU in mehrere Rechenkerne zerfallen, die über einen ECU Bus kommunizieren können. Das SPES Metamodell [HHK10] lässt auch für die TA eine rekursive Dekomposition zu.

In diesem die Modellierung abschließenden Abschnitt gehen wir nun auf die horizontale Beziehung der SPES Matrix ein. Diese lässt sich mit den Stichworten *Allokation* und *Deployment* beschreiben.

5.1 Allokation von Funktionen

Die FA modelliert sämtliche Systemfunktionen und ihre Beziehungen zueinander. In voller Schönheit würden die atomaren Blattfunktionen auch mit einer syntaktischen Schnittstelle und einer Verhaltensbeschreibung ausgestattet.

Die LA modelliert die gesamte Softwareapplikation und muss hierbei insbesondere die definierten Funktionen erbringen einschließlich der durch Querbeziehungen definierten Koordinierungsaufgabe.

Folglich stehen die Funktionen im FA Baum mit den Komponenten der LA in enger Beziehung. Am Beispiel der Motorsteuerung ist diese Beziehung relativ einfach: jede Basisregelfunktion ist als Unterkomponente in der Sollwertberechnung enthalten, wobei von den alternativen Funktionen des Volllastbetriebs eine Variante ausgewählt wurde. Die Modusselektion und die Multiplexerkomponente realisieren dabei die Querbeziehungen im FA Baum. Es wird also der jeweilige Betriebsmodus ermittelt und der entsprechende Regler nach außen durchgeschaltet. Die Koordinationsfunktion wird also durch das Zusammenspiel mehrerer Komponenten realisiert, während die Basisregler eins zu eins Eingang in die LA fanden.

Im Allgemeinen ist die Relation zwischen FA und LA eine n:m Beziehung, d.h. eine Funktion kann durch mehrere Komponenten gemeinsam erbracht werden, während eine Komponente einen Beitrag zu mehr als einer Funktion liefern kann. Der methodische Umgang mit dieser Allgemeinheit und insbesondere die Frage, wie weit ein Funktionsdekomposition durchgeführt werden soll, sind Gegenstand weiterer Forschung im Rahmen von SPES ZP-AP-1.

5.2 Deployment von Komponenten

Die LA beschreibt die vollständige Softwareapplikation, welche durch die Hardware der TA ausgeführt werden muss. Zu diesem Zweck wird in AutoFocus 3ein Deployment modelliert, in dem die Komponenten der LA den Steuergeräten der TA zugewiesen werden. Weiterhin werden die logischen Signale auf Sensorik, Aktuatorik und Netzwerkcontroller deployed. Sensoren können hierbei nur mit Inputports, Aktuatoren nur mit Outputports und Controller mit beiden verknüpft werden.

Das Deploymentmodell stellt also die Integration von Hardware- und Softwaremodell dar. Es kann für die weitestgehend vollständige Codegenerierung verwendet werden.

5.3 Verbau von ECUs

Nicht im Detail dargelegt wurde die vierte Spalte der SPES Matrix: die Geometrische Architektur. Sie beschreibt die Maße und Positionen von Recheneinheiten und Netzwerkverbindung wie es in entsprechenden Werkzeugen aus dem Bereich Maschinenbau bekannt ist. Hierbei interessiert uns insbesondere der Verbau der Recheneinheiten der TA, da wir so eine vollständige Nachverfolgbarkeit von Funktionen bis zur Verbauposition im Endprodukt erreichen. Wir können also den Weg, welche Funktion durch welche Komponente erbracht wird, welche Komponente wo ausgeführt wird und welche Ausführungseinheit wo verbaut ist, nachverfolgen und bspw. für Sicherheitsanalysen heranziehen.

6 Abschließende Bemerkungen

Das in diesem Durchstichprototypen vorgestellte Modell der Motorsteuerung in AutoFocus 3illustriert die Anwendung der SPES Modellierungsmatrix auf ein industrielles Szenario. Wir haben in diesem Dokument bewusst den Fokus auf die Strukturmodelle und deren Aufteilung in die einzelnen Entwicklungsperspektiven gelegt. Besonders wichtig war uns hierbei die Bedeutung der vertikalen und der horizontalen Beziehung der Element der Matrix.

In unseren Betrachtungen haben wir folgende Aspekte ausser Acht gelassen:

- Verhaltenspezifikationen, wie Automaten, Filter, Basisoperationen oder konfigurierbare PID-Regler, wurden werder in der FA noch der LA beleuchtet.
- Simulationsmodell mit interaktiver Simulation in Form von Bedienkonsolen.
- Codegenerierung und die Ausgestaltung der TA Elemente wurde nicht weiter vertieft, obwohl das Deploymentmodell als Basis für eine vollständige Codegenerierung dienen kann.
- Das Modell der LA verwendet ein globales Zeitmodell und erzwingt damit ein synchrones Ausführen von Komponenten. Multiratensysteme und sich ausschließende Funktionen können hiermit nur unzureichend realisiert werden.

Der letzte Punkt ist hierbei Gegenstand aktueller Forschung in SPES ZP-AP-1. Für die restlichen Punkte sei abschießend nochmals auf das AutoFocus 3Bilderbuch [HHLP10] verwiesen.

References

[Bau98] Horst Bauer, editor. *Dieselmotor-Management*. Robert Bosch GmbH, 1998.

[BBR+05] Andreas Bauer, Manfred Broy, Jan Romberg, Bernhard Schätz, Peter Braun, Ulrich Freund, Núria Mata, Robert Sandner, and Dirk Ziegenbein. AutoMoDe — Notations, Methods, and Tools for Model-Based Development of Automotive Software. In *Proceedings of the SAE 2005 World Congress*, volume 1921 of *SAE Special Publications*, Detroit, MI, April 2005. Society of Automotive Engineers.

[BS01] Manfred Broy and Ketil Stølen. *Specification and Development of Interactive Systems: Focus on Streams, Interfaces, and Refinement*. Springer, 2001.

[Hei10] Peter Heidl. SPES Deliverable: Beschreibung der Fallstudie *Automotive Robert Bosch*, 2010.

[HF11] Florian Hölzl and Martin Feilkas. AutoFocus 3 - A Scientific Tool Prototype for Model-Based Development of Component-Based, Reactive, Distributed Systems. In Holger Giese, Gabor Karsai, Edward Lee, Bernhard Rumpe, and Bernhard Schätz, editors, *Model-Based Engineering of Embedded Real-Time Systems*, volume 6100 of *Lecture Notes in Computer Science*, pages 317–322. Springer Berlin / Heidelberg, 2011.

[HHK10] Alexander Harhurin, Florian Hölzl, and Thomas Kofler. SPES Deliverable D1.4.A: *SPES Metamodel*, 2010.

[HHLP10] Martin Haldenmair, Florian Hölzl, Christian Leuxner, and Birgit Penzenstadler. AutoFOCUS 3 - The Picture Book (To appear.). Technical report, Technische Universität München, 2010.