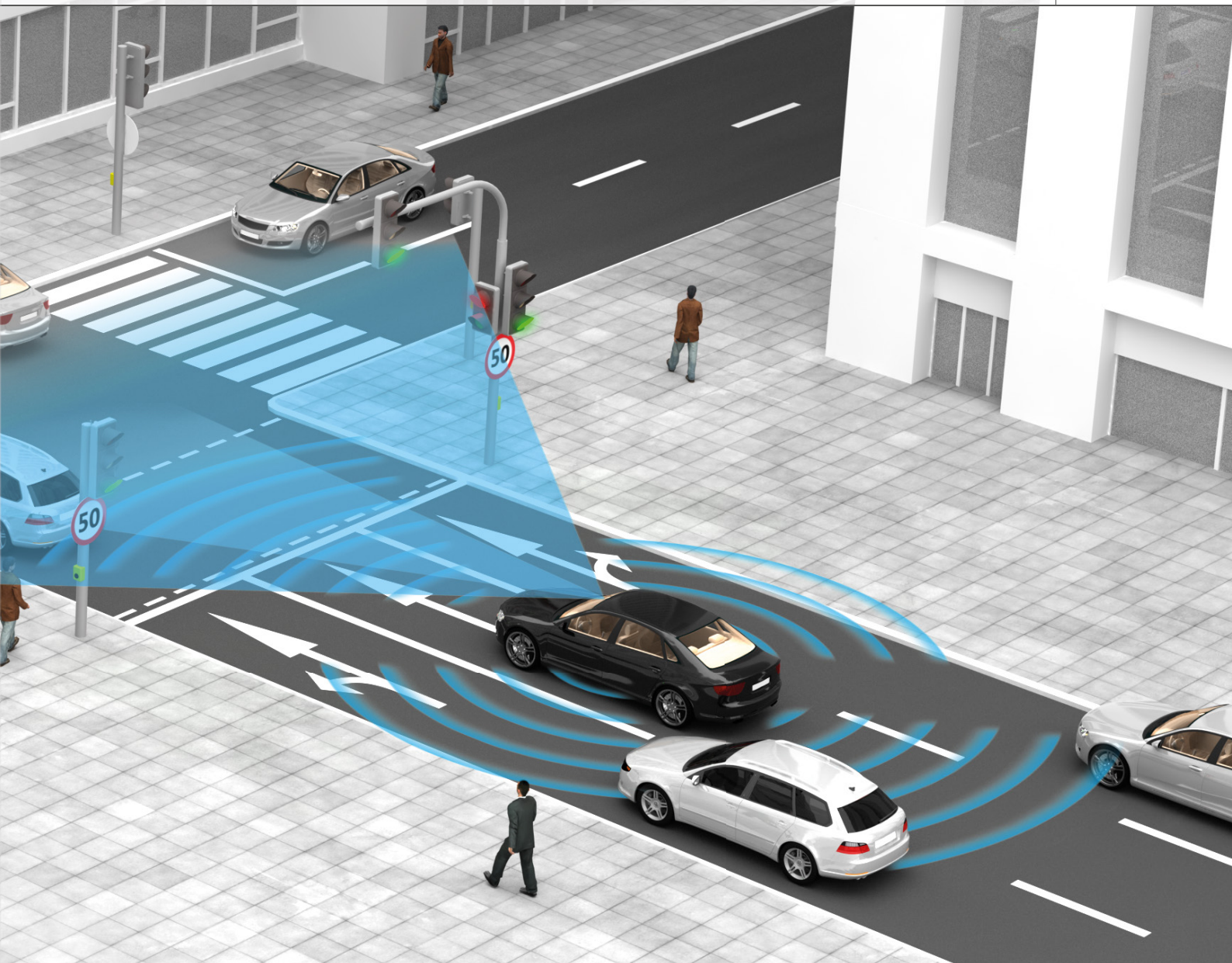


Einheitliche modellbasierte Entwicklung von Fahrerassistenzfunktionen für verschiedene Plattformen

Entwicklung intelligenter Assistenten

In einem Vorentwicklungsprojekt für sensorgestützte Fahrerassistenzsysteme bei der Automotive Safety Technologies GmbH gilt es, komplexe Daten sicher zu handhaben und zu analysieren. Der Seriercode-Generator TargetLink unterstützt diese Aufgabe und ermöglicht einen durchgängigen, effizienten Workflow.

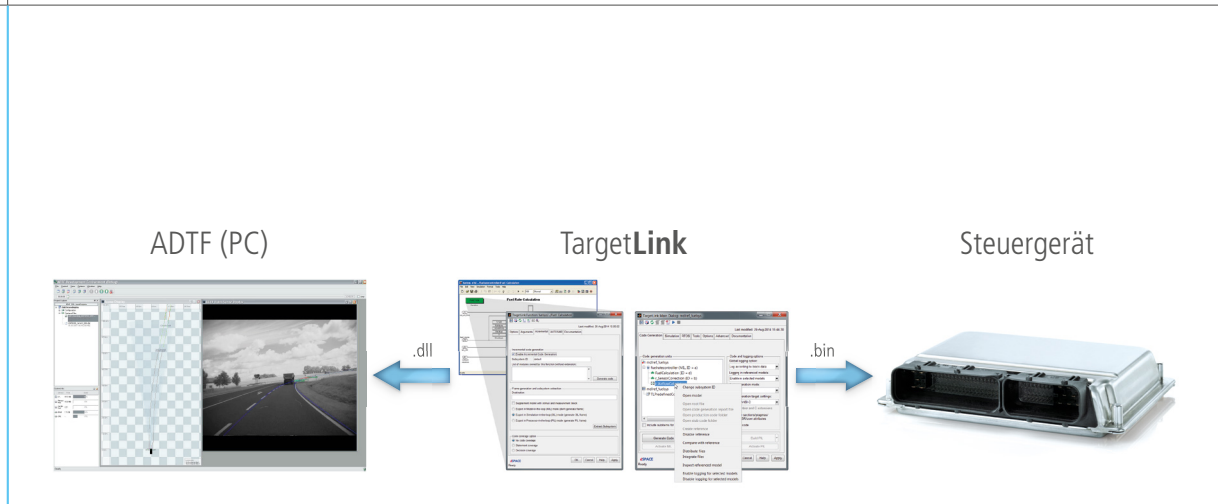


Bei Entwicklungen im Bereich der sensorgestützten Fahrerassistenz mit einer Vielzahl von Datenfusionen verschiedener Sensoren entstehen komplexe Algorithmen und Datenstrukturen sowie umfangreiche Datenmengen. In einem Vorentwicklungsprojekt mit Fokus auf Quer- und Kreuzungsverkehr geht es um die effiziente Verarbeitung von

- Datenfusionen verschiedener Sensoren (Kamera, Radar, Laser),
- Objektbildung auf Basis von Laser-Rohdaten,
- Kamerabasiertes Tracking von Objekten (z.B. Fahrzeuge, Fußgänger).

Um während der modellbasierten Entwicklung die Diagnose der komplexen Funktionen zu gewährleisten, muss eine Reihe funktionsinterner und somit nicht direkt zugänglicher Informationen zur Überprüfung und zu Testzwecken verfügbar gemacht werden. Diese Detailinformationen benötigen zusätzlichen Speicher und erhöhte Laufzeit, was nur innerhalb der Entwicklungsphase akzeptabel ist. Ab Auslieferung der Funktion werden diese Informationen nicht mehr benötigt, stattdessen liegt dann der Fokus auf minimalem Speicherbedarf und optimierter Laufzeit. Daraus ergibt sich die Notwendigkeit, eine Software-

Komponente (SWC) hinsichtlich ihres nicht-funktionalen Umfangs skalierbar zu erstellen und je nach Anwendungsfall entsprechend flexibel kompilieren zu können. Während der Entwicklungsphase stehen oft noch nicht die entsprechenden Zielplattformen zur Verfügung, bzw. ist es gerade in der Vorentwicklung wichtig, schnell prototypische Funktionen ohne Relevanz auf konkrete Plattformen umzusetzen. Hierfür eignet sich beispielsweise die Entwicklungsumgebung ADTF (Automotive Data and Time triggered Framework), die es ermöglicht, Funktionen auf PC-Basis auszu- >>



Vereinfachte Darstellung der Integration einer aus dem Funktionsmodell generierten Software-Komponente (SWC): Während im Steuergerät die SWC als Binärdatei (.bin) integriert wird, bindet ADTF die SWC als ausführbare Bibliothek (.dll, Windows) ein.

führen und mit anderen Komponenten zu verschalten.

Entwicklungsumgebung für Sensordaten

ADTF stellt eine Umgebung dar, um entweder online im Fahrzeug Sensordaten synchron aufzunehmen und der relevanten Funktion zuzuführen oder aufgenommene Daten offline abzuspielen und somit die Funktionen unabhängig vom Aufnahmezeitpunkt ablaufen zu lassen. Neue Komponenten können per Hand programmiert werden, alternativ lassen sich bereits existierende Komponenten auf der ECU ansteuern. So ist es möglich, prinzipiell

dasselbe Modell für die Steuergeräte-Code-Generierung zu verwenden sowie eine Windows®/Linux-Bibliothek zu erstellen. Zusätzlich möchte man, wie eingangs erwähnt, für die Entwicklungsphase hier möglichst viele funktionsinterne Informationen von außen einsehbar machen, um die Funktionsweise (gerade bei Fehlverhalten) analysieren und nachvollziehen zu können. Diese internen Daten müssen zu diesem Zweck allerdings bereits vom Modell zur Verfügung gestellt werden. Die Komponente, die innerhalb der Entwicklungsumgebung das kompilierte Modell aufruft, kann diese Daten dann auswerten, visualisieren oder wiederum

an andere Komponenten weiterleiten.

Anforderungen an die technische Umsetzung

Für die Umsetzung der Funktionen in ausführbare Steuergeräte-Software kommt der Seriene-Generator TargetLink® von dSPACE zum Einsatz. Mit ihm wird aus dem Funktionsmodell Code für eine SWC generiert und sowohl ins Steuergerät als auch in die Entwicklungsumgebung integriert. Für diese beiden Zielplattformen soll eine Verhaltensanalyse der SWC zur Laufzeit nachvollziehbar bzw. visualisierbar sein. Dies geschieht in beiden Laufzeitumgebungen (Entwicklungsumgebung und Steuergerät) jedoch unterschiedlich. Beim Steuergerät stellt die SWC die vordefinierten Laufzeitvariablen über ein Mess- und Kalibrierprotokoll (XCP) einem externen Werkzeug zur Verfügung. Bei der Entwicklungsumgebung hingegen können beliebige SWC-Variablen an andere Programme übergeben und ggf. visualisiert werden. Da eine PC-basierte Umgebung anders als ein Steuergerät keine Speicherbedarfsanforderung an die SWC stellt, kann die SWC beliebig um weitere Debug-Variablen erweitert werden, um die Debugging-Funktionalitäten voll auszuschöpfen. Die Herausforderung hierbei ist, einerseits den Steuergeräte-Code weitestgehend unverändert zu lassen, andererseits

Spezifikationen im TargetLink Data Dictionary, mit denen die gewünschte Flexibilität bei der Code-Generierung für Debug-Variablen ermöglicht wird.

Property	Values
Description	„Variable class“
Storage	default
Scope	global
ArgClass	< >
Volatile	off
Const	off
Macro	off
Alias	off
InitAtDefinition	off
RestartFunctionName	„InitPredictionVariables“ Initialisierungsfunktion
SectionName	„UAS_API“ Compiler-Schalter
TypePrefix	{ }
DeclarationStatements	{ }
UseName	on

„Mit TargetLink realisieren wir einen automatisierten, durchgängigen Workflow für die effiziente Entwicklung leistungsfähiger Fahrerassistenzfunktionen, der eine umschaltbare Code-Generierung sowohl für die Entwicklungsplattform als auch für den Zielprozessor unterstützt.“

Matthias Ibbrücker, Automotive Safety Technologies GmbH

die Anforderungen für die Integration der SWC zu berücksichtigen, z.B. möglichst viele Variablen für Debug-Zwecke zur Verfügung zu stellen. Eine weitere Herausforderung ist ein hoher Automatisierungsgrad für die Integration in der jeweiligen Zielplattform (Linux oder Windows), um manuelle Eingriffe im Arbeitsablauf zu reduzieren. Die Lösung basiert auf Anpassungen im SWC-Modell, in der SWC-Datenbank sowie mittels entsprechender Skripte.

Automatisierte Umsetzung mit TargetLink

Um diese Automatisierung zu erreichen und die Integration von TargetLink in den Entwicklungsprozess zu unterstützen, wurde der Callback-Mechanismus innerhalb von TargetLink genutzt. TargetLink-spezifische Hook-Funktionen unterstützen benutzerspezifische Anpassungen während des Kompilervorgangs. Benutzerdefinierte Anweisungen werden beim Aufruf der Hook-Funktionen automatisiert ausgeführt, was zusätzliche manuelle Anpassungen überflüssig macht. Mit TargetLink gelang es, die zusätzlichen Debug-Variablen als Arrays umzusetzen, die beim Kompilervorgang per Compiler-Schalter entweder erzeugt (Entwicklungsumgebung) oder ausgelassen werden (Steuergerät). Mittels Kompilervarianten kann derselbe Code für die unterschiedlichen Entwicklungsplattformen entwickelt und

beim Kompilervorgang auf die gewünschte Ausprägung reduziert werden. Da diese Debug-Variablen Arrays mit unterschiedlichen Breiten und Datentypen sein können, wurde eine wiederverwendbare generische Lösung in Form einer Bibliothek implementiert. Darüber hinaus war es mit Präprozessor-Anweisungen und automatisiert eingefügten Präfixen möglich, identischen C-Code für die unterschiedlichen Laufzeitumgebungen (Steuergerät, Entwicklungsumgebung für Windows oder Linux) abhängig vom Compiler-Schalter zu generieren. ■

*Matthias Ibbrücker, Automotive Safety Technologies GmbH
Mohinder Pandey*

Matthias Ibbrücker

Matthias Ibbrücker arbeitet in der Serienentwicklung von Kreuzungsassistenten bei der Automotive Safety Technologies GmbH in Gaimersheim, Deutschland.



Fazit

Um den Anforderungen an eine effiziente durchgängige Entwicklung gerecht zu werden, nutzt die Automotive Safety Technologies GmbH den Serien-code-Generator TargetLink zur zielplattformgerechten Code-Generierung. Per Compiler-Schalter lässt sich aus einem Funktionsmodell automatisiert Code generieren, der entweder mit erweiterten Debug-Variablen ausgestattet oder für den Betrieb auf Steuergeräten optimiert ist. Der Vorteil: Aus exakt demselben Funktionsmodell wird die Entwicklungsumgebung ADTF gespeist, um hier komfortabel Fahrerassistenzfunktionen zu entwickeln, den Code für Steuergeräte zu generieren und unter realen Laufzeitbedingungen zu testen.

Mohinder Pandey

Mohinder Pandey arbeitete in der Vor-entwicklung von Kreuzungsassistenten bei der Automotive Safety Technologies GmbH in Gaimersheim, Deutschland.

