Test von verteilten C2X-Applikationen

Jürgen Grossmann, Carsten Neumann, Andreas Hinnerichs,
Horst Rechner, Thomas Hecker, Ilja Radusch

{juergen.grossmann | carsten.neumann | andreas.hinnerichs |
horst.rechner | thomas.hecker | ilja.radusch}@fokus.fraunhofer.de

Abstract: Zukünftige Fahrzeuggenerationen werden, stärker noch als heute, in eine umfassende Kommunikationsinfrastruktur eingebunden sein, die den Datenaustausch sowohl zwischen Fahrzeugen wie auch mit der Verkehrsinfrastruktur ermöglicht. Auf Basis einer solchen Car to X (C2X) Infrastruktur lassen sich intelligente Telematik- und Fahrassistenzsysteme realisieren, die den Straßenverkehr insgesamt effizienter, sicherer und komfortabler machen werden. Voraussetzung dafür ist eine hohe Qualität und Zuverlässigkeit dieser Systeme.

Die systematische Qualitätssicherung solcher Systeme bringt eine Reihe neuer Herausforderungen mit sich, die in ihrer Kombination bisher einzigartig sind und besondere Anforderungen an die Beschreibung der Testfälle sowie an die für die Prüfung verwendeten Prüfsysteme stellen. In diesem Papier stellen wir eine flexible Referenzarchitektur für einen C2X-Prüfstand vor, mit dem sich C2X-Systeme effektiv und abgestimmt auf ihre spezifischen Kommunikationsanforderungen testen lassen. Ausgehend von den speziellen Prüfanforderungen der C2X-Systeme zeigen wir weiterhin, wie sich Spracherweiterungen für die standardisierte Testbeschreibungssprache TTCN-3 motivieren lassen und zeigen ihre Integration mit der zuvor definierten Prüfstandsarchitektur. Die in diesem Papier vorgestellten Ergebnisse sind im Rahmen der Projekte sim^{TD} und TEMEA entwickelt worden.

1 Motivation

C2X-Systeme kommunizieren mittels komplexer Nachrichten. Sowohl die Nachrichtenübermittlung wie auch der Nachrichteninhalt hängen vom Ort und Zustand des Fahrzeuges ab und weisen einen Komplexitätsgrad auf, der bisher im Fahrzeugbereich unüblich gewesen ist. Die Nachrichtenformate und Nachrichtenprotokolle für den europäischen Raum werden derzeit von der ETSI [ETS09a] standardisiert. Grundsätzlich wird zwischen sogenannten Cooperative Awareness Messages (CAM) und sog. Decentralized Environmental Notification Messages (DENM) unterschieden. Während CAMs von den Fahrzeugen regelmäßig versandt werden und den aktuellen Fahrzeugstatus an die Umgebung propagieren, sind DENMs ereignisbezogene Nachrichten, die andere Verkehrsteilnehmer über konkrete Ereignisse (z.B. Hindernisse, Baustellen, etc.) informieren sollen.

Der komplexe Zusammenhang zwischen dem dynamischen Fahrzeugstatus, der Nachrichtenverteilung und dem Nachrichteninhalt führt dazu, dass C2X-Applikationen i.d.R. schwer zu prüfen sind. Neben der Stimulation des Systems mit Nachrichten muss zusätzlich

und kontinuierlich der Fahrzeugstatus (d.h. Geschwindigkeit, Beschleunigung, Position etc.) gesetzt werden.

2 Lösungsansatz

Um der Komplexität der Anwendungen gerecht zu werden, wurden in der Vergangenheit für die Prüfung von C2X-Systemen häufig Verkehrssimulatoren [SMR08, EOSK05, GM09] eingesetzt. Diese wurden benutzt, um die notwendigen Positions- und Geschwindigkeitsdaten für die Fahrzeuge zu liefern. Während sich die überwiegende Anzahl dieser Ansätze auf die Prüfung der C2X-Software beschränken, adressieren wir in diesem Artikel analog zu [SGR+09] den Test vollständig integrierter C2X-Systeme, bestehend aus der Applikationssoftware, der Basissoftware sowie der Zielhardware. Um zusätzlich bei der Auswertung der Tests einen hohen Grad an Automation gewährleisten zu können und die Nachvollziehbarkeit, Dokumentation und Wiederholbarkeit der Tests sicherzustellen, setzen wir für die Spezifikation der Tests auf die formale Sprache TTCN-3 [ETS09b] bzw. TTCN-3 embedded [SG08, GSW08, TEM09a]. TTCN-3 embedded wurde im Rahmen des Projekts TEMEA [TEM09b] entwickelt und bietet den vollen Umfang von TTCN-3, d.h. die Unterstützung für die Prüfung kommunikationsbasierter, ereignisgesteuerter Systeme. Darüber hinaus stellt es Erweiterungen für den Test kontinuierlicher, d.h. auf Sensordaten basierender Steuer- und Regelsysteme, zur Verfügung.

Im Folgenden werden wir die Definition einer flexiblen Referenzarchitektur für einen C2X-Prüfstand vorstellen und zeigen, wie sich C2X-Systeme auf einem solchen Prüfstand durch die Verwendung von TTCN-3 embedded systematisch und effektiv testen lassen.

3 Referenzarchitektur für einen C2X-Prüfstand

Am Fraunhofer Institut FOKUS wird eine Prüfstandsarchitektur entwickelt, die es erlaubt, sowohl einzelne C2X-Applikationen wie auch die Kommunikation zwischen verschiedenen C2X-Systemen zu testen, bevor die Systeme in reale Fahrzeuge verbaut werden. Abbildung 1 zeigt eine solche Architektur, die als Grundlage für den Prüfstandsaufbau in den Projekten sim^{TD} [sim] und PreDrive [PRE] verwendet wurde.

Grundsätzich besteht ein solcher Prüfstand aus einem Prüfstandsrechner und dem zu testenden System inklusive zusätzlicher Instrumentierung[1]. Das zu testende System kann, abhängig vom Testziel, aus einer oder mehreren On Board Units (OBUs) und Infrastrukturkomponenten, wie z.B. Road Side Units (RSUs), bestehen. Die *Testing API* auf dem Test-PC bietet eine gemeinsame Schnittstelle zum Einspielen aller Stimulationsdaten in die angeschlossenen OBUs und RSUs sowie zur Analyse des Systemverhaltens. Über diese Schnittstellen können verschiedene Programme, Testwerkzeuge wie auch Simulatoren in das Testsystem integriert werden. Die Gegenstellen der Testing API auf den OBUs und

[1] Die zum Prüfstand gehörenden Komponenten sind in der Abbildung fett hervorgehoben.

RSUs sind die *C2XAPI* und die *Vehicle API (VAPI)* [EK04]. Die C2XAPI gehört zum Testsystem und wird ausschließlich zu Testzwecken auf die Zielsysteme installiert. Sie dient dazu, C2X-Nachrichten, die über die Testing API in das Testsystem eingespielt werden, an die internen Schnittstellen des C2X-Systems zu übermitteln. Die Übermittlung einer Nachricht erfolgt entweder an den *COM Client*, der die Nachricht über die Kommunikationsschnittstelle versendet oder an die *Local Data Map (LDM)*, die eine Nachricht für die lokale Verwendung innerhalb einer OBU verfügbar macht.

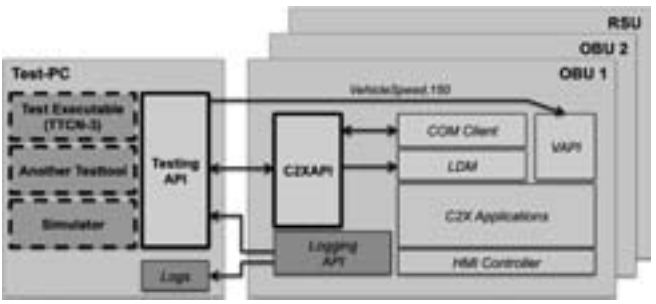

Abbildung 1: Architektur des Prüfstands

Während die C2XAPI für die Verarbeitung der C2X-Nachrichten zuständig ist, wird die VAPI verwendet, um den aktuellen Fahrzeugstatus zu setzen. Im Gegensatz zu den C2X-Nachrichten, die i.d.R. sporadisch gesendet werden, setzt sich der Fahrzeugstatus aus sich kontinuierlich ändernden Größen (z.B. Geschwindikeit, Längsbeschleunigung, Querbeschleunigung etc.) zusammen. Um ein interaktives Testen zu ermöglichen, ist zusätzlich ein Rückkanal vorgesehen, der über den COM Client eingehende und ausgehende Nachrichten sowie Präsentationsaufträge vom *HMI Controller* (letzteres über Verwendung der *Logging API*) an die Testing API zurückmeldet.

Der Prüfstandsrechner ist grundsätzlich in der Lage, mehrere OBUs bzw. RSUs über Ethernet mit synthetischen Eingabedaten zu stimulieren. Die Testing API bietet in diesem Zusammenhang eine werkzeugunabhängige Schnittstelle, die eine Integration mehrerer Testwerkzeuge ermöglicht und die Verteilung der Daten an das System unter Test übernimmt. Die über die Testing API ins System eingebrachten Daten werden von den Fahrzeugfunktionen (*C2X Applications* in der Abbildung) wie reale Daten behandelt und erlauben so einen systematischen Test der integrierten C2X-Funktionen auf dem Prüfstand.

Als Ergebnis des Tests werden sämtliche Aktionen und Daten der Fahrzeugfunktionen, LDM, COM Client, HMI Controllerund gegebenenfalls VAPI geloggt. Die automatische Auswertung dieser Logs kann auf dem Test-PC über TTCN-3 geschehen.

4 TTCN-3 embedded

TTCN-3 embedded ist eine Weiterentwicklung des bereits in der Telekommunikationsbranche etablierten Standards TTCN-3. Die neuen TTCN-3 embedded Konzepte erlau-

ben die Spezifikation von Tests, die kontinuierlich physikalische Größen wie Geschwindigkeit, Beschleunigung, Temperatur etc. in ihrer Abhängigkeit von der Zeit adressieren können. Zu diesem Zweck werden indizierte Datenstromports zur Verfügung gestellt, die einen wahlfreien Zugriff auf die aktuellen Werte sowie die Historie eines Signalverlaufs ermöglichen. Zusätzlich erlauben angepasste Kontrollstrukturen die Spezifikation getakteter, zyklischer Zuweisungen. Listing 1 zeigt einen TTCN-3 embedded Programmabschnitt, der die Geschwindigkeit eines Fahrzeugs in Abhängigkeit von der Ausführungszeit kontinuierlich erhöht (Zeile 2). Zur selben Zeit wird geprüft, ob die Umdrehungszahl des Motors unter einer gegebenen Schwelle (hier 5000.0 U/min, Zeile 3) verbleibt. Stimulation und Prüfung erfolgen in einer zeitgesteuerten Schleife (Zeilen 1,4), deren Schrittweite zuvor separat festgelegt werden kann. Der Programmabschnitt wird verlassen, wenn die Geschwindigkeit den Wert von 180.0 Km/h erreicht.

Listing 1: Konzepte zur Beschreibung und Auswertung kontinuierlicher Signale

```
cont{                                                              1
 VelocityStream.value:= 10.0 * duration;                           2
  assert(EngineSpeed.value <= 5000.0)}                             3
until (VelocityStream.value >= 180.0)                              4
setverdict(pass);                                                  5
```

Neben physikalischen Größen lassen sich mit TTCN-3 embedded nach wie vor die klassischen Eigenschaften kommunikationsbasierter Systeme prüfen. Zu diesem Zweck steht der volle Funktionsumfang des aktuellen TTCN-3 Standards zur Verfügung, u.a. ein mächtiges Typsystem zur Spezifikation von Nachrichtenformaten sowie die Möglichkeit Musterausdrücke (Templates) zu definieren, mit denen sich eingehende Nachrichten flexibel auf ihren Inhalt prüfen lassen. Listing 2 zeigt die Spezifikation eines Musterausdrucks, der eingehende Nachrichten filtert und nur Nachrichten durchlässt, deren Feld `_messagekind` mit dem Wert `'CAM'` belegt ist. Das heißt DENM Nachrichten werden unterdrückt.

Listing 2: Definition der Nachrichtenformate

```
template C2XMessage CAMMessage :={ _messageKind:="CAM",            1
                                   _timestamp:=?, _pload:=?}       2
```

5 Der Test von C2X-Systemen mit TTCN-3 embedded

Betrachten wir ein vereinfachtes Szenario aus der Fahrzeug-zu-Fahrzeug Kommunikation. Ein mit Kommunikationstechnologie ausgerüstetes Fahrzeug sendet bei einer gegebenen Toleranz von 10% zyklisch alle 5 Sekunden seinen Status per CAM an andere Fahrzeuge. Ändert sich hingegen die Geschwindigkeit des Fahrzeugs mit einem Wert größer $2m/s^2$, wird sofort eine Statusmeldung abgesetzt. Wir prüfen, ob bei sich ändernden Geschwindigkeitswerten die Nachricht ad-hoc verschickt wird.

CAMs werden in TTCN-3 durch den Datentyp C2XMessage modelliert und haben eine Nachrichtenkennung (`_messageKind`), die sie als CAM Nachricht (siehe Listing 2) identifizieren. Wir wählen einen Prüfaufbau, der der Referenzarchitektur in Abbildung 1 ent-

spricht und aus zwei OBUs besteht. Eine für den Prüfling (OBU1) und eine als Empfänger (OBU2), sprich als Bestandteil des Testsystems.

Listing 3: CAM Test

```
testcase PostionMessageTest_AdHocCAM runs on TTCN3Tester{                    1
                                                                             2
  cont{OBU1_VAPI_LongAccel.value:= 0.4;}                                     3
  until{                                                                     4
   [duration> 4.5 and duration < 5.5]                                        5
    OBU2_COM_IN.receive(posMessage){repeat}                                  6
   [] OBU2_COM_IN.receive(CAMMessage){setverdict(fail); stop}                7
   [] OBU2_COM_IN.receive(?){continue}                                       8
   [] [now >= 30.0]{}}                                                       9
                                                                             10
 cont{OBU1_VAPI_LongAccel.value:= 2.5;}                                      11
 until{                                                                      12
   [] OBU2_COM_IN.receive(CAMMessage){setverdict(pass)}                      13
   [] OBU2_COM_IN.receive(?){continue}                                       14
   [duration > 0.5] {setverdict (fail)}}                                     15
}                                                                            16
```

Das Setzen der Größen für den Fahrzeugstatus im Prüfling erfolgt über die VAPI und wird im Gegensatz zur CAM Nachricht als kontinuierlicher Stimulus modelliert. Das Testszenario in Listing 3 ist zweistufig.

Im ersten Abschnitt des Tests (Zeilen 3 - 9) wird die Normalsituation, d.h. das Verhalten bei einer nahezu gleichförmigen Geschwindigkeit, geprüft. Das Fahrzeug wird durch Beschleunigungsdaten stimuliert, die gemäß einer zuvor gesetzten Samplingrate zyklisch gesetzt wird (Zeile 3). Am COM Client eingehende Nachrichten werden entsprechend ihres Nachrichteninhalts gefiltert. Zeitkorrekte zyklische CAM-Nachrichten sowie beliebige DENM-Nachrichten werden geschluckt (Zeile 5,6 sowie Zeile 8). Werden azyklische CAM-Nachrichten empfangen, so schlägt der Testfall fehl (Zeile 7). Nach einer Dauer von 30 Sekunden (Zeile 9) wechselt der Testfall in den zweiten Abschnitt (Zeile 11-15), in dem für eine halbe Sekunde auf die Ankunft einer Ad-hoc CAM-Nachricht gewartet wird. Wird keine CAM-Nachricht empfangen, schlägt auch hier der Testfall fehl.

6 Zusammenfassung

In diesem Kurzartikel haben wir die grundlegenden Herausforderungen für das Testen interaktiver Fahrzeugsysteme beschrieben, den Aufbau eines Prüfstandes zum Testen von C2X-Systemen und den Einsatz von TTCN-3 embedded als Spezifikationssprache für die formale Definition ausführbarer Tests skizziert. Die dargestellte Prüfstandsarchitektur erlaubt den systematischen Test von C2X-Systemen und ist flexibel auf verschiedene Prüfszenarien anpassbar (Test einzelner C2X-Applikationen auf einer OBU, Test der C2X-Kommunikation, Test eines C2X-Systems über mehrere OBUs). Die Verwendung von TTCN-3 erlaubt die Automatisierung der Tests und sorgt zusätzlich für die bei sicherheitskritischen Systemen notwendige Formalisierung und Wiederholbarkeit der Tests. Diese Arbeit wurde im Rahmen des Projektes sim^{TD} durch die Bundesministerien für Wirtschaft

und Technologie (BMWi) und Bildung und Forschung (BMBF) gefördert und durch das Bundesministerium für Verkehr, Bau und Stadtentwicklung (BMVBS) unterstützt sowie im Projekt TEMEA durch Mittel finanziert, die aus dem Europäischen Fonds für Regionale Entwicklung (EFRE) stammen.

Literatur

[EK04] Oliver Eggers and Jens Krüger. Das Fahrzeug als standardisiertes, generisches Objektmodell, in:. In *ÏMA 2004 - Informationssysteme für mobile Anwendungen", Beiträge zum gleichnamigen 2. Braunschweiger Symposium am 20. und 21. Oktober 2004*, pages 205 –218, Braunschweig, 2004.

[EOSK05] Stephan Eichler, Benedikt Ostermaier, Christoph Schroth, and Timo Kosch. Simulation of Car-to-Car Messaging: Analyzing the Impact on Road Traffic. In *MASCOTS '05: Proceedings of the 13th IEEE International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems*, pages 507–510, Washington, DC, USA, 2005. IEEE Computer Society.

[ETS09a] ETSI. ITS Activity Report 2009; http://portal.etsi.org/its/ActivityReport2009.asp, 2009.

[ETS09b] ETSI: ES 201 873-1 V4.1.1. Methods for Testing and Specification (MTS). The Testing and Test Control Notation Version 3, Part 1: TTCN-3 Core Language, Febr. 2009.

[GM09] Markus Glaab and Alois Mauthofer. NEW TEST AND EVALUATION METHODS FOR FUTURE CAR2X COMMUNICATION BASED DRIVER ASSISTANCE. In *21st Int. Technical Conference on the Enhanced Safety of Vehicles (ESV)*, Stuttgart, 2009.

[GSW08] Jürgen Großmann, Ina Schieferdecker, and Hans-Werner Wiesbrock. Modeling Property Based Stream Templates with TTCN-3. In Kenji Suzuki, Teruo Higashino, Andreas Ulrich, and Toru Hasegawa, editors, *TestCom/FATES*, volume 5047 of *Lecture Notes in Computer Science*, pages 70–85. Springer, 2008.

[PRE] PRE-DRIVE C2X. http://www.pre-drive-c2x.eu/index.dhtml/394bd0148054d449805e/-/deDE/-/CS/-/. Die Website wurde zuletzt am 22.04.2010 besucht.

[SG08] Ina Schieferdecker and Jürgen Großmann. Testing hybrid control systems with TTCN-3: an overview on continuous TTCN-3. *STTT*, 10(4):383–400, 2008.

[SGR+09] Oliver Sander, Benjamin Glas, Christoph Roth, Jurgen Becker, and K. D. Muller-Glaser. Testing of an FPGA Based C2X-Communication Prototype with a Model Based Traffic Generation. In *RSP '09: Proc. of the 2009 IEEE/IFIP Int. Symposium on Rapid System Prototyping*, pages 68–71, Washington, DC, USA, 2009. IEEE Computer Society.

[sim] sim^{TD} : Sichere Intelligente Mobilität - Testfeld Deutschland. http://www.simtd.de. Die Website wurde zuletzt am 20.04.2010 besucht.

[SMR08] Björn Schünemann, Kay Massow, and Ilja Radusch. A Novel Approach for Realistic Emulation of Vehicle-2-X Communication Applications. In *VTC Spring*, pages 2709–2713. IEEE, 2008.

[TEM09a] TEMEA. Draft Proposal for Continuous TTCN-3 Concepts; http://t-ort.etsi.org/file_download.php?file_id=2282&type=bug, 2009.

[TEM09b] TEMEA. TEMEA Project (Testing Methods for Embedded Systems of the Automotive Industry), founded by the European Community (EFRE), http://www.temea.org, 2009.