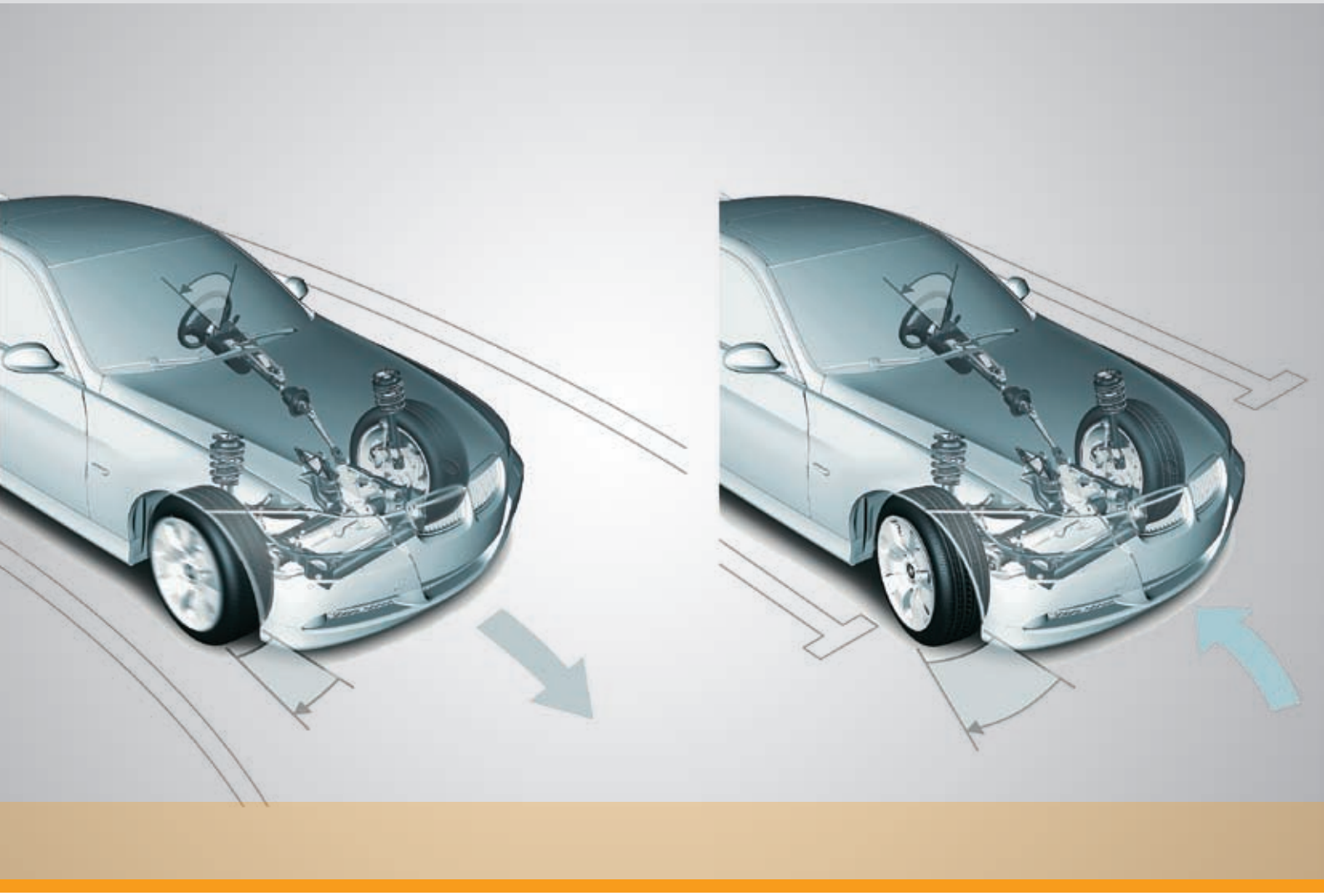

# Laufzeitanalysen zur frühzeitigen Absicherung von Software

Es ist erstaunlich, wie viele Embedded Systeme entwickelt werden, ohne das Laufzeitverhalten der Software im Auge zu behalten. Durch fehlerhaftes Timing verursachte Probleme sind meist schwierig zu erkennen und noch schwieriger zu lösen. Nicht so bei der Entwicklung der Aktivlenkung von BMW, bei der von Anfang an das Laufzeitverhalten in das Softwaredesign einbezogen und mit Hilfe der Symtavison GmbH und der Gliwa GmbH fortlaufend durch Messungen und Analysen abgesichert wurde.

## 1 Einleitung

Mit der Einführung der Aktivlenkung des 5er-BMW beschritt BMW technologisches Neuland. Erstmals wurde in einem Großserienprojekt der elektronisch gesteuerte Eingriff in die Lenkung zugelassen, um erhöhte Agilität und neue fahrdynamische Sicherheitsfunktionen zu ermöglichen. Eine der Maßnahmen zur Erreichung der geforderten Sicherheit war und ist ein stabiles und vorhersagbares zeitliches Verhalten der Software. Ähnlich dem Design der Funktionalität der Software („was passiert?“) soll das zeitliche Verhalten genau definiert sein („wann passiert es?“). Das klingt zunächst trivial, kann doch durch entsprechende Konfiguration des Betriebssystems vorgegeben werden, wann welcher Anteil der Software gerechnet werden soll. Die interessante Frage in diesem Zusammenhang lautet aber: kann in allen Fällen sichergestellt werden, dass immer genug Rechenleistung zur Verfügung steht, um die Vorgabe einzuhalten?

## 2 Technische und kommerzielle Aspekte

Mit der Beantwortung dieser und weiterer die Laufzeit betreffender Fragen wurden externe Experten beauftragt. Bereits seit der ersten Aktivlenkungs-Generation stellt die Gliwa GmbH die Messtechnik zur Erfassung, und das Tool „traceGURU“ zur Visualisierung und Analyse des Laufzeitverhaltens bereit. Beides spielt bei der Software-Fehlersuche und Laufzeitoptimierung eine zentrale Rolle. In regelmäßigen Abständen werden Laufzeitanalysen durchgeführt, um eventuelle Probleme frühzeitig zu erkennen und um den Laufzeitbedarf der Software über den Projektverlauf hinweg zu dokumentieren. Die „traceGURU“-Reports sind mittlerweile fester Bestandteil der Freigabedokumente der BMW-Integrationsstufen. Zudem ermöglicht das Verfahren die Vorhersage der zukünftig benötigten Rechenleistung. Bei Projektbeginn der dritten Generation der Aktivlenkung zeigten zum Beispiel Laufzeituntersuchungen des Vorgängerprojektes und der geplanten zusätzlichen Funktionen, dass einer der beiden Prozessoren des Systems durch ein langsameres und wesentlich günstigeres (flashloses) Derivat ersetzt werden könnte. Die Vorhersage der Prozessorauslastung des Systems traf auf drei Prozent genau, und die Entscheidung für den günstigeren Prozessor erweis sich als richtig.

Solche Vorhersagen werden durch den konsequenten Einsatz von Tools zur Messung und Analyse ermöglicht – der Trend vom Beschreiben und Verstehen hin zum Kontrollieren ist deutlich erkennbar. In der dritten Generation der Aktivlenkung wurde bei BMW daher eine weitere Stufe der Absicherung eingeführt, die so genannte Scheduling-Analyse zum mathematischen Nachweis von Laufzeitanforderungen. Zum Einsatz kommt das Scheduling-Analyse-Werkzeug „SymTA/S“ der Symtavision GmbH. SymTA/S findet und überprüft zielgerichtet die in der Realität seltenen, aber kritischen „Timing Corner Cases“ – und zwar auch dann, wenn diese in den realen Messungen gar nicht aufgetreten sind. Das klingt zunächst paradox, wird jedoch ermöglicht durch die mathematische Analyse der zeitlichen Eigenschaften des Systems, die in Abschnitt 3 erläutert wird. Automatisierte Reports dokumentieren die Ergebnisse in der gewohnten Form mittels Tabellen und Timing-Diagrammen – die Entwickler kommen mit der grundlegenden Mathematik nicht in Berührung.

Durch die offenen Schnittstellen für die Eingangsdaten von SymTA/S gestaltet sich der Einsatz einfach. So werden zum Beispiel zunächst am Testplatz oder im Fahrzeug „traceGURU“-Messungen durchgeführt, deren Ergebnisse in XML-Dateien exportiert und wiederum von SymTA/S eingelesen und analysiert werden. Das Tool prüft systematisch sämtliche „Timing Corner Cases“, Im besten Fall gibt das System „grünes Licht”. , weist aber bei Problemen gezielt auf die Ursache hin. All dies geschieht vollkommen automatisiert in die Testumgebung integriert als fester Bestandteil der Systemverifikation, **Bild 1**. Neben den verschiedenen technischen und kommerziellen Aspekten, die dafür sprechen, die Laufzeit von Embedded-Systemen nicht aus den Augen zu verlieren, gibt es noch einen weiteren Grund, der in den letzten Jahren zunehmend an Gewicht gewonnen hat. Die Produkthaftung macht es geradezu erforderlich, bei der Absicherung von Embedded-Systemen nicht am falschen Ende zu sparen.

## Die Autoren


**Dr. Marek Jersak**
ist Mit-Gründer und CEO der Symtavision GmbH in Braunschweig.

**Dr. Kai Richter**
ist Mit-Gründer und CTO der Symtavision GmbH in Braunschweig.

**Hans Sarnowski**
ist Softwareentwickler der Aktivlenkung bei der BMW AG in München.

**Peter Gliwa**
ist Geschäftsführer der Gliwa GmbH in München.

## 3 Laufzeitmessung

Die Funktionsweise von „traceGURU“ lässt sich in vielerlei Hinsicht mit der eines Speicheroszilloskopes vergleichen. Jedoch werden betriebssystembezogene Größen aufgezeichnet, nämlich die Zeitpunkte für Aktivierung, Start und Ende von Tasks, Interrupts und Prozessen/Runnables. Aus diesen Daten lässt sich die Laufzeitsituation im aufgezeichneten Zeitraum rekonstruieren, grafisch darstellen und analysieren.

**Bild 2** gibt eine solche Situation wider: über eine Zeitachse sind die Zustände der Tasks und Interrupts aufgetragen. Eine hellgraue Einfärbung repräsentiert den Zustand „ready“, eine dunkelgraue den Zustand „running“. Abläufe, Prozessorauslastung, Verteilung et cetera lassen sich „qualitativ“ auf einen Blick erkennen.

Beispielsweise zeigt Bild 1 den zeitlichen Bezug der SPI-Interrupts zu den Tasks. In der Darstellung sind neben den Tasks und Interrupts noch Userevents (grüne Striche) und eine Stopwatch (blauer Balken) zu erkennen. Sie erlauben dem Anwender, benutzerdefinierte Daten einschließlich des zeitlichen Bezugs aufzuzeichnen respektive beliebige Codeabschnitte auszumessen.

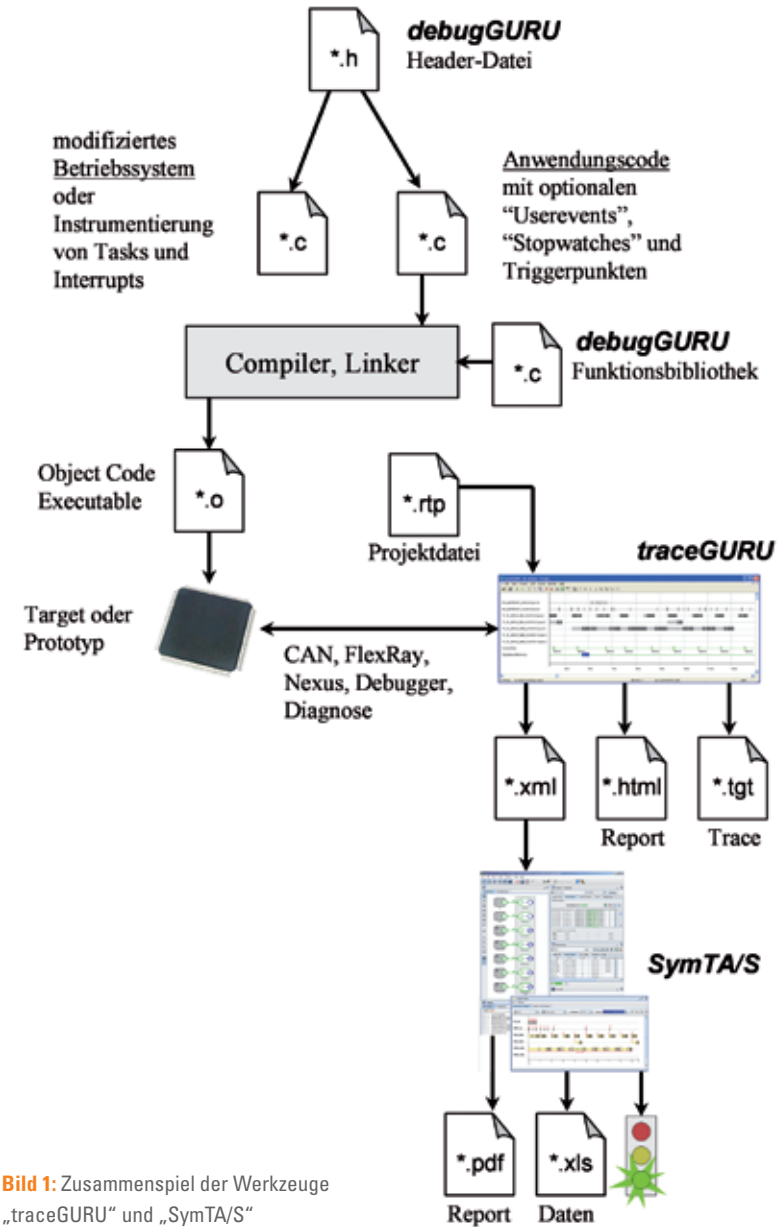


**Bild 1:** Zusammenspiel der Werkzeuge „traceGURU“ und „SymTA/S“

Eine genaue „quantitative“ Auswertung kann unter anderem über die Generierung eines Reports erfolgen, der neben Werten für die gemessene „Worst case execution time“ (WCET) der einzelnen Tasks, Prozesse/Runnables und Interrupts viele weitere Timinggrößen dokumentiert. Dabei werden sowohl Brutto- als auch Nettozeiten festgehalten, bei letzteren wird die Dauer von eventuell stattgefundenen Unterbrechungen heraus gerechnet.

Nachfolgeprojekte werden auf die von der Gliwa GmbH vollkommnen neu entwickelte Messsoftware „T1“ setzen. Die Auswertung erfolgt mit T1 direkt auf dem Steuergerät in Echtzeit und die Ergebnisse, also zum Beispiel die gemessenen WCETs von Tasks, können bei Bedarf über die Hardwareschnittstelle abgerufen werden. Entscheidend ist bei dieser targetseitigen „Online-Analyse“, dass lückenlos alle Interrupts und Tasks berücksichtigt werden und trotzdem die erforderliche Bandbreite zur Außenwelt frei wählbar ist – es werden ja nur die Ergebnisse übermittelt und das kann sehr langsam erfolgen.

Optional lassen sich diese Ergebnisse noch im nichtflüchtigen Speicher ablegen. So kann zum Beispiel bei einer Erprobung ohne Mehraufwand die Absicherung der Laufzeit nebenher erfolgen, eine Interaktion mit der Messtechnik ist dafür nicht erforderlich. Dem Entwickler entgeht nicht ein einziger Laufzeitausreißer.

Die Online-Analyse beeinträchtigt die Laufzeit des Zielsystems nur unwesentlich, da sie im Vergleich zu konkurrierenden Ansätzen von der Messwerterfassung entkoppelt ist.

Die Aufzeichnung der Laufzeitdaten erfolgt abhängig vom eingesetzten Betriebssystem entweder über die Modifikation des Betriebssystems oder über die Instrumentierung der Tasks und „Interrupts“. Bei „OSEK“- und Autosar-Betriebssystemen bietet sich hier ein Überladen der „TASK(...)“- und „ISR (...)“-Makros an. Prinzipiell kann aber jedes Betriebssystem ausgemessen werden, das die jeweiligen Zustände „suspended“, „ready“ und „running“ kennt oder vergleichbare Zustände bietet. Für die Übermittlung der zur Laufzeit aufgezeichneten Daten stehen die Hardwareschnittstellen „Debugger“, „Nexus“ und „CAN“ zur Verfügung, ab Frühjahr 2009 auch die Anbindung über Diagnose und Flexray.

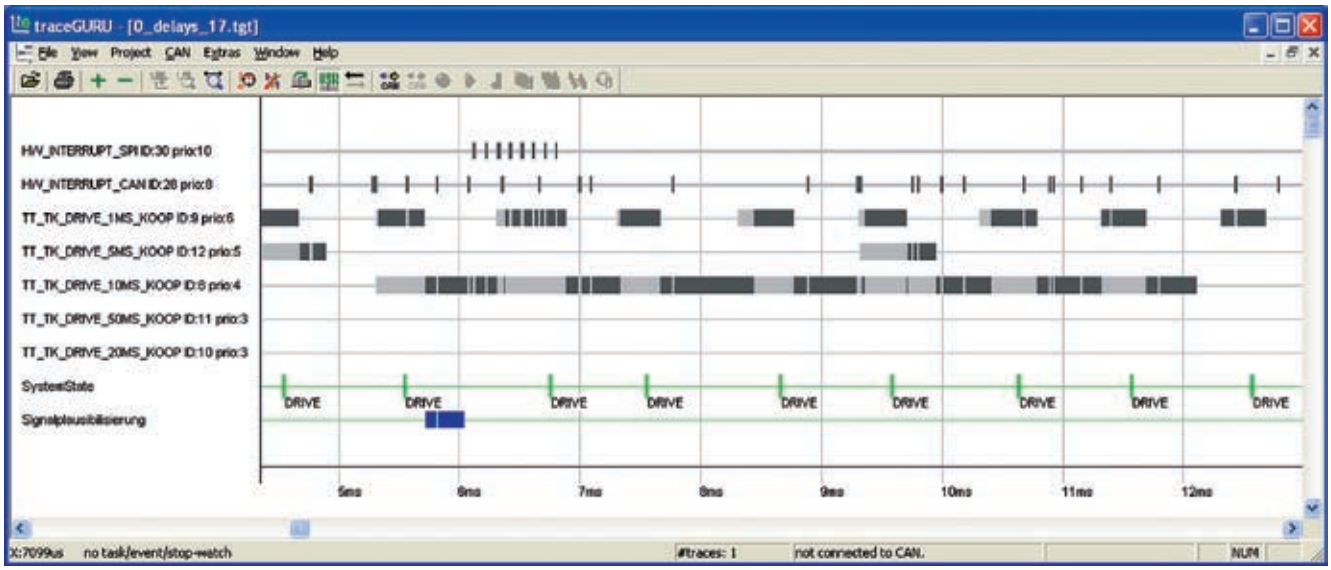


**Bild 2:** „traceGURU“ als „Oszilloskop des Betriebssystems“ „SymTA/S“ nach erfolgter Scheduling-Analyse

## 4 Scheduling-Analyse

Scheduling-Analysen gehen im Vergleich zur Messung einen entscheidenden Schritt weiter, indem sie gezielt kritische Engpässe und Fehlersituationen aufdecken, auch wenn diese beim Messen gar nicht aufgetreten sind. Während die beim Messen erzeugten „Traces“ eingefrorene (das heißt statische) Einzelsituationen darstellen, berücksichtigt das Werkzeug „SymTA/S“ die in der Realität variierenden dynamischen Größen explizit als Variablen. Hierzu zählen insbesondere dynamische Interrupts oder Events sowie Schwankungen in der Task-Laufzeit. Noch komplexer können sich Task-Abhängigkeiten auswirken, die mit Hilfe von so genannten „Event Models” systematisch erfasst werden.

Die eigentliche Analyse erfolgt in zwei Schritten. „SymTA/S“ zerlegt die importierten Traces in ihre Einzelbestandteile, um sie dann – unter systematischer Berücksichtigung der dynamischen Variablen – wieder zusammenzubauen. Somit erreicht „SymTA/S“ zielgenau diejenigen kritischen Situationen, in denen Last und Laufzeiten maximal werden, **Bild 3**.

Dies soll am Beispiel der „10ms-Task“ verdeutlicht werden, die während der Trace-Dauer von maximal vier CAN-Interrupts unterbrochen wurde. In anderen Situationen während der Trace-Dauer wurden hingegen deutlich mehr CAN-Interrupts beobachtet, die jedoch zum Teil zwischen zwei Instanzen der „10ms Task“ lagen. „SymTA/S“ legt nun diese Einzelbestandteile automatisch so zusammen, dass die gegenseitigen Störungen maximal werden. Das Timing-Diagramm in **Bild 4** zeigt, dass die „10ms-Task“ tatsächlich von bis zu zehn „CAN- Interrupts“ unterbrochen werden könnte. Dieser Fall wurde während der Trace-Dauer nicht beobachtet, ist jedoch nicht ausgeschlossen.

Solche kritischen Randfälle systematisch aufzuzeigen erhöht die Testabdeckung von Steuergeräten signifikant. Denn erst wenn gezeigt wird, dass Zeitbudgets beziehungsweise Deadlines auch in diesen kritischen Fällen nicht verletzt werden, kann von einer zuverlässigen Funktion in allen Situationen ausgegangen werden Die notwendige Genauigkeit erreicht „SymTA/S“ durch speziell angepasste Analysebibliotheken. Das Portfolio

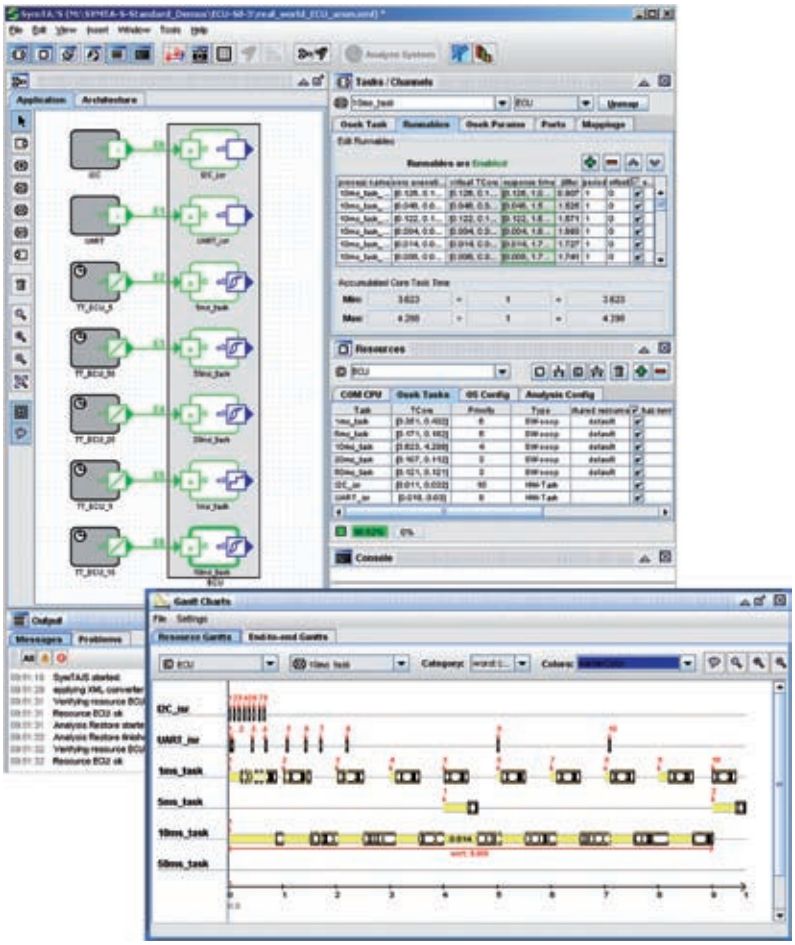

**Bild 3:** „SymTA/S“ nach erfolgter Scheduling-Analyse

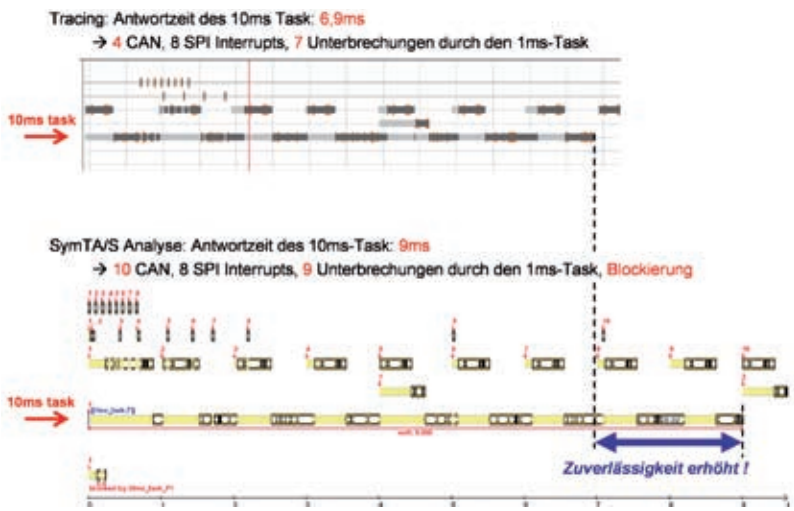


**Bild 4:** Der Vergleich von Messung und „SymTA/S“-Analyse zeigt die Erhöhung der Testabdeckung

umfasst verschiedene OSEK-Varianten, Autosar-OS, sowie Module für CAN und Flexray. Diese sind sowohl einzeln lauffähig, als auch zur Systemanalyse nach dem Baukastenprinzip kombinierbar. Entwickler beginnen immer mehr das Timing eines Systems noch während oder sogar vor der Entwicklung der Steuergerätesoftware an virtuellen Echtzeit-Prototypen zu analysieren – zu einem Zeitpunkt, zu dem noch keine Traces zur Verfügung stehen. Hier zeigt sich eine weitere Stärke von Scheduling-Analyse, die nicht auf ausführbaren Code angewiesen ist. Vorhersagen, welchen Einfluss längere oder kürzere Tasks oder der Prozessortakt auf Zeitverhalten und Performance haben (“Was-wäre-wenn“-Analysen) sind ebenso möglich wie automatisierbare Optimierungen. Daraus leiten sich bereits in sehr frühen Phasen der Plattform- und Architekturentwicklung sowie Softwareintegration wichtige Entwicklungsvorgaben ab, zum Beispiel Software-Budgets und Hardware-Dimensionierungen.

## 5 Fazit

Am Beispiel der Aktivlenkung bei BMW wird deutlich, welche Vorteile für Zuverlässigkeit und Kostenoptimierung der systematische Einsatz von Tools für die Timing-Analyse in der Steuergeräteentwicklung über mehrere Produktgenerationen hinweg bringt. Das Thema Timing ist in der Automobilindustrie heute noch jung, gewinnt aber schnell an Bedeutung. Belege sind die aktuellen Arbeiten des Timing-Teams in Autosar unter Beteiligung von Symtavision, sowie die parallel laufenden Europäische Projekte „TIMMO“ und „LL-TIMES“. Für die Industrie geht es um die Beherrschung der Echtzeitfähigkeit, Zuverlässigkeit und Sicherheit unter Kostengesichtspunkten bei zunehmender Systemkomplexität. Entscheidungen im Bereich E/E-Architektur, Software-Architektur oder Funktionsintegration hängen hiervon ab, um Risiken neuer Assistenzfunktionen und Antriebeskonzepte zu minimieren. Die systematische Integration des Themas Timing in den Entwicklungsprozess wird mit für den Erfolg entscheidend sein. ■

ATZlive

**29. UND 30. JANUAR 2009 | MÜNCHEN | MTZ KONFERENZ - MOTOR**

# Alternative Antriebe für Automobile

## Internationale Konferenz

**Eine Veranstaltung in Zusammenarbeit mit**

- Hybridkonfiguration
- Brennstoffzellen
- Zukünftige Verbrennungsmotoren
- Alternative Energieträger

**ATZlive**
**Abraham-Lincoln-Straße 46**
**65189 Wiesbaden | Deutschland**

**Telefon +49(0)611. 7878-131**
**Telefax +49(0)611. 7878-452**
**atzlive@springer.com**

**PROGRAMM UND ANMELDUNG**

**www.ATZlive.de**