Automatisierung und Kreativität

Ist die Grenze beherrschbarer Komplexität erreicht?

Dr. Robert Hilbrich

1 Einführung

Der technologische Fortschritt unserer Gesellschaft spiegelt sich in der hohen Leistungsfähigkeit unserer Fahrzeuge wider. Piloten müssen ihre tonnenschweren Flugzeuge heute nur noch indirekt steuern, in dem sie über einen kleinen Joystick mit dem Steuerungssystem interagieren. Die Komplexität der richtigen Ansteuerung von Höhen-, Seiten- und Querruder bei gleichzeitiger Anpassung der Schubkraft tritt fast vollkommen in den Hintergrund. Autofahrer können sich durch eine Vielzahl von Assistenzsystemen bei der Fahrt unterstützen lassen. Unliebsame Aufgaben können sie an den *Autobahn-Piloten*, den *Stau*-Assistenten oder den *Park*-Assistenten delegieren. Der Schritt zu voll-automatischen Fahrzeugen ist nicht mehr weit.

Der *Computer* in seiner Rolle als automatisiertes Steuersystem nimmt dem Menschen immer mehr Aufgaben ab. Dies betrifft insbesondere Aufgaben von denen unsere persönliche Sicherheit abhängt. Ein kleiner Fehler im Steuerungssystem eines Flugzeugs – zum Beispiel ein kleiner Zeitversatz zwischen der Ansteuerung des linken und rechten Höhenruders – kann bereits genügen, um das Flugzeug in einen aerodynamisch instabilen Zustand zu bringen. Im Auto können Ungenauigkeiten bei der Spur-Erkennung eines *Autobahn-Piloten* schnell katastrophale Folgen haben.

Trotz dieser Risiken übergeben wir die Verantwortung für unser Wohlergehen immer stärker an Computersysteme und vertrauen auf deren Zuverlässigkeit. Die Statistik der Aus- und Unfälle belegt, dass diese Entscheidung rückblickend richtig war. In Anbetracht der sprunghaften technologischen Entwicklungen in den letzten Jahren stellt sich die Frage, ob dieses Vertrauen auch *zukünftig* gerechtfertigt ist. Sind unsere Entwicklungsmethoden und -werkzeuge auch zukünftig „mächtig“ genug, um die Komplexität der Entwicklung derartiger Systeme zu beherrschen?

Die Steuersysteme im Auto und Flugzeug bestehen aus vielen vernetzten Computersystemen. Sie interagieren mit zahlreichen Sensoren und Aktuatoren, um ihren Aufgaben gerecht zu werden. Ein vollwertiges „Fly-by-Wire“ Steuersystem setzt sich aus ca. 40 vernetzten Computersystemen zusammen. Moderne Fahrzeuge beinhalten nicht selten

mehr als 75 Computersysteme, um die Vielzahl an Funktionen zu realisieren. Die Entwicklung dieser Systeme gerät in Anbetracht dieser Komplexität an ihre Grenzen. Sie ist gefordert, alle Anforderungen zu erfüllen und dabei eine maximale Zuverlässigkeit bei minimalen Kosten zu erzielen.

Diese Komplexität in der Entwicklung ist ohne den Einsatz von Computer nicht mehr zu bewältigen. Obwohl sich ihr Einsatzspektrum in der Vergangenheit deutlich erweitert hat, ist die grundlegende Arbeitsteilung zwischen dem menschlichen Entwickler und seinem elektronischen Assistenten noch immer unverändert. Während der Computer die wiederkehrenden Routine- und Analyseaufgaben übernimmt, sind die kreativ-konstruktiven Tätigkeiten noch immer dem Menschen überlassen. Der Mensch erschafft mit seiner Arbeit etwas Neues und Originelles – zum Beispiel die Architektur einer neuen Flugsteuerung oder einen Algorithmus für eine neue Assistenzfunktion. Die Transformation dieser „Artefakte“ in Maschinencode und deren Analyse auf Programmierfehler wird durch den Computer durchgeführt. Aufgrund dieser Arbeitsteilung ist die Komplexität der zu entwickelnden Systeme durch die menschliche Verarbeitungskapazität begrenzt, denn jedes Entwicklungsvorhaben ist zunächst auf die Durchführung von kreativ-konstruktiven Tätigkeiten angewiesen.

Grundsätzlich muss der Entwickler dazu die Anforderungen an das System verinnerlichen und mit den Eigenschaften und Fähigkeiten der zur Verfügung stehenden „Lösungsbausteine“ in Übereinkunft bringen. Eine vollständige und fehlerlose Berücksichtigung *aller* Anforderungen bei der Systementwicklung übersteigt bereits heute in vielen Fällen die Verarbeitungskapazität der Entwickler. Spätestens mit der Nutzung von Mehrkernprozessoren, die eine parallele Ausführung verschiedene Softwaremodule zur gleichen Zeit ermöglichen, ist die Grenze der sicher beherrschbaren Komplexität überschritten. Eine a priori Abschätzung *aller* Auswirkungen dieser Parallelität überschreitet das Vermögen menschlicher Entwickler.

Die gestiegene Komplexität der Systeme führt nicht nur die kreativ-konstruktiven Tätigkeiten der Entwicklung an ihre Grenzen. Auch die automatisierten Verfahren zum Testen der entwickelten Systeme sind diesem Anstieg nicht mehr vollumfänglich gewachsen. Der Nachweis über die *vollständige* Korrektheit eines derartigen Systeme unter *allen* Randbedingungen lässt sich nicht mehr allein durch die Beobachtung von dessen Verhalten erbringen. Aufgrund der Vielzahl der dann zu prüfenden System- und Umgebungszustände ist ein derartiges Vorgehen nicht länger ökonomisch sinnvoll.

ALT

Bei der Entwicklung von komplexen sicherheitskritischen Systemen, zum Beispiel der Steuerungssysteme an Bord eines modernen Flugzeugs, sehen sich Entwickler mittlerweile mit der Herausforderung konfrontiert, auch kreativ-konstruktive Tätigkeiten automatisieren zu müssen. Dies ist einerseits die Folge einer stark gestiegenen Komplexität in den funktionalen Anforderungen der Systeme in Verbindung mit einer exponentiell gestiegenen Komplexität der technologischen „Bausteine“, die für die Synthese zur Verfügung stehen.

Andererseits ist der Drang zur Automatisierung der kreativ-konstruktiven Tätigkeiten auch eine Folge der hohen Anforderungen an die funktionale Korrektheit und Zuverlässigkeit dieser Systeme.

In der Praxis lässt sich die vollständige Korrektheit eines derartigen Systeme mittlerweile nicht mehr allein durch die extrinsische Beobachtung und Analyse von dessen Verhalten überprüfen. Aufgrund der Vielzahl der zu prüfenden Systemzustände ist ein derartiges Vorgehen zum vollständigen Analyse in der Praxis nicht länger ökonomisch sinnvoll. Daher muss die Korrektheit eines Systems verstärkt auf der Grundlage seines *Konstruktionsprozesses* nachgewiesen werden. Die Automatisierung des Konstruktionsprozesses kritischer Artefakte ist daher eine zentrale Herausforderung für die ingenieurwissenschaftliche Forschung in diesem Bereich, um die Weiterentwicklung dieser Klasse von Systemen zu ermöglichen und zugleich auch zukünftig deren Zuverlässigkeit und Korrektheit zu gewährleisten.

DISSERTATION

Die zugrunde liegende Dissertation befasst sich mit der automatisierten Konstruktion eines zentralen Artefakts sicherheitskritischer Systeme: der Zuweisung von Ressourcen, zum Beispiel der Rechenzeit auf einem Prozessor oder einem Teil des Arbeitsspeichers, zu den einzelnen Softwarekomponenten – oder anders formuliert: der Platzierung der Softwarekomponenten auf den Ressourcen des Systems. Diese Zuweisung ist scheinbar unspektakulär, doch dieser Eindruck trügt. Eine „richtige“ und „ausreichende“ Zuweisung von Ressourcen, ist für das Gesamtsystem *essentiell*. Sie ist die Grundvoraussetzung dafür, dass zeitkritische Steuerungsmodule, zum Beispiel das Airbagsteuergerät im Auto, zum „richtigen“ Zeitpunkt über „genügend“ Ressourcen zur Abarbeitung ihrer Steuerungslogiken verfügen.

Mit der Einführung von modernen „Mehrkern-“ Prozessoren, die eine parallele Ausführung von Softwarekomponenten erlauben, wurde die Möglichkeit zu einer starken Funktionsverdichtung in der Architektur von sicherheitskritischen Systemen geschaffen. Ein Mehrkern-Prozessor kann – im Gegensatz zu einem klassischen Ein-Kern-Prozessor – verschiedene Softwarekomponenten zur gleichen Zeit ausführen, so dass Geräte eingespart und damit die Effizienz des Systems gesteigert werden können. Die Ressourcenzuweisung ist damit nicht nur eine Grundvoraussetzung für die korrekte Funktionsweise des Systems, sondern zugleich auch der zentrale Ansatzpunkt für Optimierungen zur Steigerung der Ressourceneffizienz.

Die Synthese einer solchen Zuweisung erfordert eine vollständige Koordination aller Ressourcenzugriffe der Softwarekomponenten bereits zum Zeitpunkt der Entwicklung. Ihre Erstellung ist äußerst aufwändig und fehlerträchtig, denn sie ist sowohl mit der Komplexität der Anforderungen *aller* Softwarekomponenten als auch der Komplexität *aller* Ressourcen der Hardwareplattform konfrontiert. Während die Ressourcenzuweisung bei realen Systemen in der Vergangenheit noch manuell konstruiert werden konnte, ist spätestens mit der Einführung von Mehrkern-Prozessoren ein Komplexitätsniveau erreicht, bei dem eine Fortsetzung tradierter Vorgehensweisen nicht länger zielführend ist.

Vor dem Hintergrund dieser Herausforderung wird in der zugrunde liegenden Dissertation ein automatisiertes Verfahren zur Konstruktion einer solchen Ressourcenzuweisung für Mehrkernprozessoren entwickelt und anhand von zwei Fallbeispielen für komplexe Systeme aus der Luft- und Raumfahrt erprobt.

Mittlerweile wird dieses Verfahren in leicht angepasster Form bei einem Hersteller ziviler Flugzeuge für die Optimierung der Systemarchitektur eines Flugzeugtyps eingesetzt. Durch die Verwendung dieses Verfahrens konnte die benötigte Zeitdauer für die Konstruktion einer vollständigen und fehlerfreien Zuweisung von ca. 12-15 Monaten auf einen Zeitraum von etwa 5 Minuten reduziert werden. Die Kürze der Konstruktionsdauer ermöglichte erstmals die Synthese einer Vielzahl unterschiedlicher Zuweisungen, so dass erstmalig auch Entwurfsalternativen zur Optimierung verglichen und letztendlich Geräte eingespart werden konnten.

In den folgenden Abschnitten ...

2 Ergebnisse der Dissertation

Hier kommt der Inhalt der Dissertation etwas stärker im Detail; dazu die Ergebnisse ...

3 Gesellschaftliche Bedeutung

- Zunächst kann damit die Entwicklung von Systemen beschleunigt werden
- Systeme können optimiert werden, so dass weniger Ressourcen benötigt werden
- Weniger Fehler im System
- ...

Aber, diese Arbeit ist ein guter Gegenstand, um Fragen nach der Beherrschbarkeit heutiger Technologien zu stellen und zu diskutieren

- Was sind die Nachteile die sich daraus ergeben?
- Ist es nicht sinnvoll, wenn sich Menschen so lange mit den Anforderungen beschäftigen?
- Sollten wir Dinge konstruieren und nutzen, deren Aufbau das Verständnis des Menschen übersteigt?
- ...

4 Zusammenfassung