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 häufiger 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 jedoch 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 sicher zu beherrschen?

Die Steuersysteme im Auto und Flugzeug bestehen aus vielen vernetzten Computern. Sie interagieren mit zahlreichen Sensoren und Aktuatoren, um ihre Aufgaben zu erfüllen. Ein vollwertiges „Fly-by-Wire“ Steuersystem für Verkehrsflugzeuge setzt sich aus ca. 40 vernetzten Computern zusammen. Moderne Fahrzeuge beinhalten nicht selten mehr als 75 Computer, um die Vielzahl an Funktionen zu realisieren. Die Entwicklung derartiger

Steuersysteme gerät in Anbetracht dieser Komplexität an ihre Grenzen. Sie ist gefordert, alle Funktionen unter Berücksichtigung verschiedener denkbarer Umgebungsbedingungen und -zustände zu realisieren und dabei eine maximale Zuverlässigkeit bei minimalen Kosten zu erzielen.

Diese Komplexität ist ohne den Einsatz von Computer in der Entwicklung nicht mehr zu bewältigen. Obwohl sich das Einsatzspektrum der Computer in der Systementwicklung in den letzten Jahren 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, bleiben die *kreativ-konstruktiven Tätigkeiten* 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. Erst die Transformation dieser neuen „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 der Entwickler kann nur eine begrenzte Zahl an Anforderungen an ein neu zu entwickelndes System verarbeiten und diese auch nur mit einer begrenzten Zahl zur Verfügung stehender „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* ist die Grenze der sicher beherrschbaren Komplexität überschritten. Während klassische *Einkernprozessoren* die Ausführung von nur *einem* Softwaremodul zu jedem Zeitpunkt ermöglichten und damit auch nur eine kleine Anzahl an Softwaremodulen pro Prozessor integriert werden konnte, bieten *Mehrkernprozessoren* aufgrund ihrer deutlich gestiegenen Rechenleistung mehr Flexibilität. Mit Hilfe ihrer unabhängig arbeitenden Rechenkerne ermöglichen sie erstmals eine *parallele Ausführung* verschiedener Softwaremodule zur gleichen Zeit, so dass wesentlich mehr Softwaremodule auf einem Prozessor integeriert werden können. Mehrkernprozessoren mit über 1000 parallel arbeitenden Rechenkernen sind bereits frei verfügbar. Sie versprechen eine sehr hohe Rechenleistung bei einem äußerst geringen Energieverbrauch und stellen damit vielversprechende „Lösungsbausteine“ für die Steuersysteme der Zukunft dar. Allerdings überschreitet eine präzise und vollständige a priori Abschätzung *aller* Auswirkungen dieser Verdichtung von Softwaremodulen auf einem Mehrkernprozessor und der daraus resultierenden Ressourcenkonkurrenz das Vermögen menschlicher Entwickler. Sie können die Zuverlässigkeit eines Systementwurfs auf Basis von Mehrkernprozessoren nur noch sehr eingeschränkt beurteilen.

Derart hochkomplexe Systeme führen nicht nur die menschliche Entwicklungsarbeit an ihre Grenzen. Auch die automatisierten Verfahren zum Testen der Neuentwicklungen sind dieser hohen Komplexität nicht mehr vollumfänglich gewachsen. Wurde bisher die Korrektheit eines Systems durch die extrinsische Beobachtung seines funktionalen Verhaltens getestet, genügt dies heute nicht mehr. Die Zeit, die dann für die Prüfung aller

System- und Umgebungszustände benötigt werden würde, übersteigt die vorgesehene Entwicklungszeit häufig um ein Vielfaches. Daher lässt sich der Nachweis über die *vollständige* Korrektheit eines Steuersystems unter *allen* Randbedingungen nicht mehr allein durch die Beobachtung von dessen Verhalten erbringen.

Dies bedeutet, mit Blick auf die gewachsenen Ansprüche an die Funktionalität von Steuersystemen ist die Grenze der Mächtigkeit der etablierten Entwicklungsmethoden und -werkzeuge erreicht. Ihre Weiterentwicklung ist die Voraussetzung, um auch zukünftig neue Steuersysteme mit einem erweiterten Funktionsumfang bei gleichzeitiger Aufrechterhaltung einer hohen Zuverlässigkeit zu konstruieren.

Ein vielversprechender Ansatz zur Lösung dieser Problematik liegt in der *Automatisierung* der kreativ-konstruktiven Tätigkeiten. Dies bricht mit der traditionellen Arbeitsteilung in der Entwicklung. Bei diesem Ansatz besteht die Aufgabe des menschlichen Entwicklers darin, die Anforderungen an das zu entwickelnde Steuersystem vollständig, präzise und fehlerfrei zu beschreiben. Im Anschluss konstruiert der Computer unter Zuhilfenahme der verfügbaren Lösungsbausteine automatisiert ein geeignetes „Artefakt“, das alle gestellten Anforderungen erfüllt.

Vorteile dieses Ansatzes liegen zum einen darin, dass Computer in der Lage sind, die gestellten Anforderungen *vollständiger* und *präziser* mit den technischen Möglichkeiten in Übereinkunft zu bringen. Die Grenze der beherrschbaren Komplexität lässt sich damit nach oben verschieben. Zum anderen eröffnet die Automatisierung der Konstruktion die Möglichkeit, die Korrektheit eines Systems auf der Grundlage seines Konstruktionsprozesses zu begründen anstatt dafür eine unvollständige Beobachtung des Systemverhaltens zu verwenden. Daher ist die Automatisierung der kreativ-konstruktiven Tätigkeiten eine zentrale Herausforderung für die ingenieurwissenschaftliche Forschung im Bereich der Systementwicklung, um die Weiterentwicklung von Steuersystemen zu ermöglichen und auch zukünftig deren Zuverlässigkeit zu gewährleisten.

Die zugrunde liegende Dissertation befasst sich mit der automatisierten Konstruktion eines zentralen „Artefakts“ eines Steuersystems:

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