

Kurzfassung

Industriewaschmaschinen sind Waschmaschinen, die vorwiegend in Wäschereien, Heimen oder Krankenhäusern eingesetzt werden. Sie sind größer, vielseitiger und können im Gegensatz zu konventionellen Haushaltswaschmaschinen weitgehend frei programmiert werden. Die Automatisierung dieser Industriewaschmaschinen kann durch den Einsatz von Mikrocontrollern und der zugehörigen Steuerungssoftware erfolgen. Da Software zu einem wichtigen Bestandteil eingebetteter Systeme geworden ist, muß sie eine bestimmte, an ihr Einsatzgebiet gebundene Qualität aufweisen. Formale Methoden stellen eine mögliche Entwicklungsmethodik dar, um Sicherheits- und Qualitätseigenschaften von Software zu verbessern. Der folgende Beitrag beschreibt Erfahrungen bei der formalen Spezifikation der Steuerungssoftware einer Industrie-waschmaschine.

1. Einleitung

Eine Waschmaschine stellt ein klassisches Automatisierungssystem mit typischen Regelungs- und Steuerungsaufgaben dar. Der Wettbewerb auf dem Markt und neue technische Entwicklungen haben in den letzten Jahren zu einer Verbesserung der Leistungsmerkmale, der Bedienbarkeit und der Qualität derartiger Automatisierungssysteme geführt.

Viele Automatisierungssysteme haben Produktcharakter, d.h. sie werden oft in größeren Stückzahlen hergestellt. Hier spielt die Qualität der Software eine besondere Rolle. Einerseits kann es sein, daß diese Software Funktionen mit Sicherheitsverantwortung übernehmen muß, andererseits können durch

Softwarefehler verursachte Folgekosten, z. B. durch Rückrufaktionen, durchaus ein Unternehmen in finanzielle Not bringen.

Ziel des vorgestellten Projektes war es, praktische Erfahrungen beim Einsatz formaler Methoden - insbesondere des nachfolgend vorgestellten Werkzeuges VSE - zur Entwicklung von Automatisierungssystemen zu sammeln und die Einsatzfähigkeit auf einer breiten Basis für ein in sich abgeschlossenes und recht komplexes Softwaresystem zu prüfen.

Im Folgenden werden zuerst das technische System, seine Architektur und dann die Anforderungen an die Steuerungssoftware präsentiert. Anschließend wird das Vorgehen, die verwendete Methodik und das für die formale Spezifikation verwendete Werkzeug kurz vorgestellt. Zuletzt werden die positiven und negativen Erfahrungen, die bei der Modellierung gemacht wurden, beschrieben.

2. Systembeschreibung

2.1. Das technische System

Das technische System Industriewaschmaschine besteht aus einer Waschtrommel, einem internen und einem externen Tank, mehreren Ein- und Auslassventilen, einer Abwasserpumpe und vier Pumpen für Flüssigwaschmittel. Die Anordnung ist schematisch und vereinfacht in Abbildung 1 dargestellt.

Zur Ermittlung des aktuellen Betriebszustands dienen ein Wärmesensor, ein Unwuchtsensor und ein diskreter Wasserstandssensor. Die Aktorik umfaßt einen Motor für den Waschvorgang, einen Motor für das Schleudern, eine Heizung und ein elektromagnetisches Schloß für das Entriegeln der Trommeltür. Die Industriewaschmaschine ist ca. 2 m hoch, 1 m breit und 1.5 m lang. Das Fassungsvermögen beträgt 16 kg Trockenwäsche.

Die Kopplung der Maschine mit der Steuerungshardware erfolgt über Relais (Abbildung 2). Die Aktoren können nur angesteuert werden, wenn der Steuerrechner und die Waschmaschine selbst die Betriebsbereitschaft mit einem Signal (`m_bereit`, `s_bereit`) anzeigen. Fehlt eines dieser Signale, wird die Waschmaschine in einen sicheren Zustand gebracht. Sicherer Zustand heißt,

daß Motoren, Pumpen und die Heizung abgeschaltet werden und die Trommeltür verriegelt bleibt.

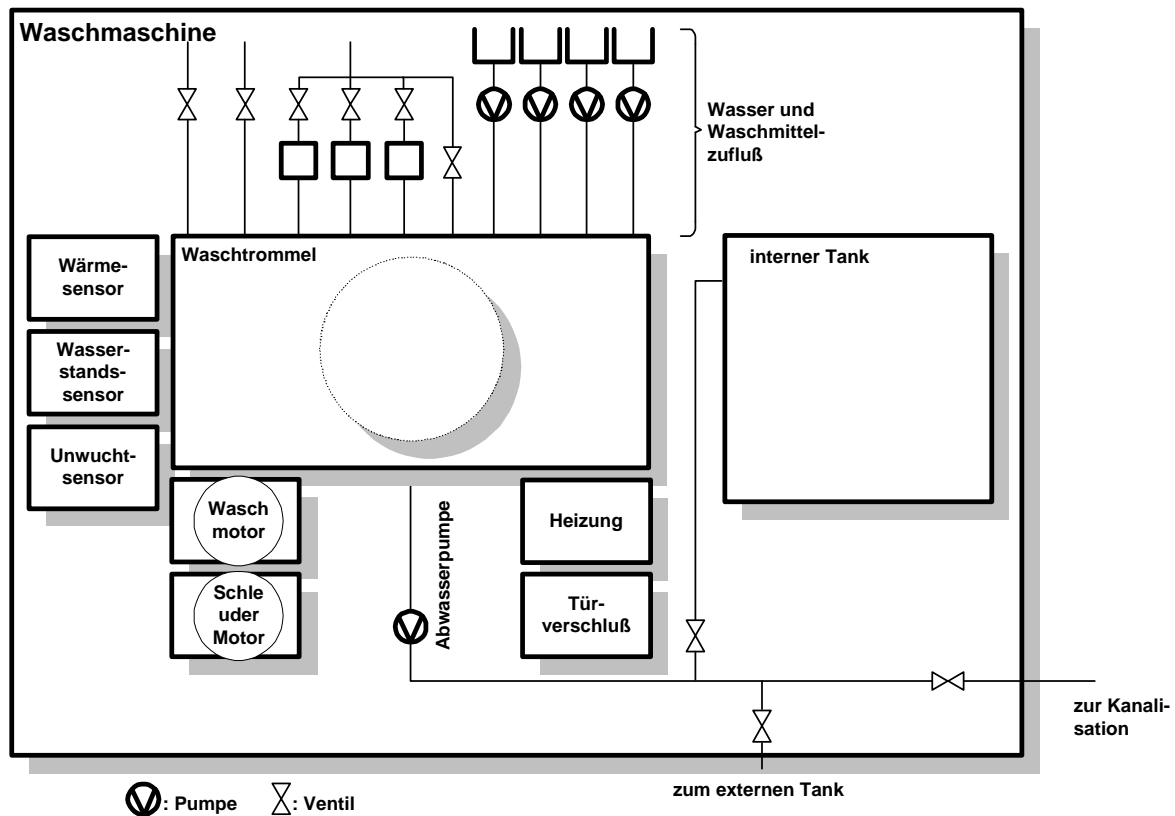


Abbildung 1: Die Waschmaschine als technisches System

An den Steuerrechner ist ein Bedienpanel und ein Chipkartenlesegerät angeschlossen. Über dieses Panel kann der Bediener den Waschvorgang starten und abbrechen. Zusätzlich können von einem besonders dazu autorisierten Benutzer Waschprogramme zusammengestellt, von einer Chipkarte gelesen oder auf eine Chipkarte geschrieben werden.

2.2. Funktionale und sicherheitskritische Anforderungen

Einige der funktionalen Anforderungen an die Software ergeben sich zwangsläufig aus der im vorigen Kapitel vorgestellten Architektur. Bei der Entwicklung der Steuerungssoftware für die Waschmaschine müssen zusätzlich folgende wichtige funktionale Anforderungen berücksichtigt werden:

- (F1) Die Waschprogramme sollen individuell zusammengestellt werden können.
- (F2) Die Speicherung der Waschprogramme soll auf Chipkarten vorgenommen werden.
- (F3) Die Steuerung der Waschmaschine soll aufgrund der Waschprogramme erfolgen, wobei unzulässige Programmschritte oder -kombinationen softwaremäßig verhindert werden müssen.

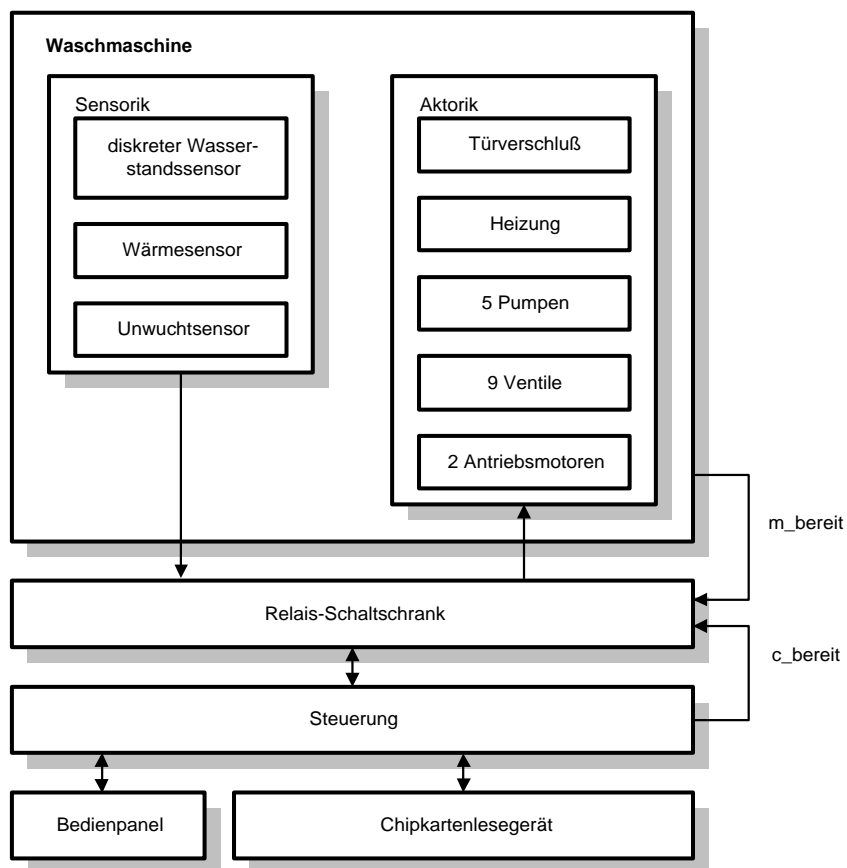


Abbildung 2: Systemarchitektur der Waschmaschine

Die zu berücksichtigenden sicherheitsrelevanten Anforderungen sind wie folgt festgelegt:

- S(1) Für die Durchführung des Aufheizens muß der Wasserstand in der Waschmaschine mindestens ein Drittel (1:3) des maximalen Wasserstands betragen.
- S(2) Vor dem Öffnen der Trommeltür darf sich kein Wasser in der Maschine befinden und alle Aktoren müssen einen definierten, für die Umwelt

ungefährlichen Zustand eingenommen haben:

1. Wasch- und Schleudermotor abgeschaltet und
2. Heizung ausgeschaltet und
3. alle Ventile geschlossen und
4. alle Pumpen abgeschaltet

S(3) Beim Schleudervorgang darf kein Wasser in der Trommel vorhanden sein.

S(4) Tritt beim Schleudern eine Unwucht auf, muß der Schleudervorgang abgebrochen werden.

3. Formale Spezifikation

3.1. Begriffsdefinition

Unter *formalen Methoden* sollen nachfolgend Methoden und Vorgehensweisen zur Spezifikation von Software verstanden werden, die auf mathematischen Modellen basieren und eine definierte Syntax und Semantik besitzen. Sie sind logikbasiert und an algebraische Spezifikationen angelehnt. Der mit mathematischen Beweisverfahren durchgeführte Nachweis von Eigenschaften wird mit dem Begriff *Verifikation* bezeichnet.

3.2. Einführung in die VSE-Methodik

Um den für die Praxis relevanten Anforderungen an ein Werkzeug zur formalen Spezifikation und Verifikation von Softwaresystemen Rechnung zu tragen, wurde im Auftrag des Bundesamts für Sicherheit in der Informationstechnik (BSI) das Werkzeug „Verification Support Environment (VSE)“ entwickelt. Seit Anfang 1996 wird VSE auch am Institut für Automatisierungs- und Softwaretechnik (IAS) der Universität Stuttgart in der Forschung und Lehre eingesetzt.

VSE umfaßt neben Verifikationswerkzeugen auch die zugehörige Entwicklungsmethodik für Systeme mit Sicherheitsverantwortung. Mit dem Werkzeug können Systeme spezifiziert, die Einhaltung sicherheitskritischer Eigenschaften mathematisch nachgewiesen und als Endergebnis Programmcode in Ada oder C generiert werden.

Die Modellierung in VSE erfolgt in drei verschiedenen Abstraktionsstufen (Abbildung 3). Auf der obersten Ebene werden sicherheitsrelevante

Anforderungen abstrakt in einem *Sicherheitsmodell* beschrieben. Dazu müssen konventionell definierte Anforderungen in die Spezifikationssprache VSE-SL umgesetzt werden. Ein Sicherheitsmodell enthält keine Aussagen über die Realisierung eines Systems. Für die Steuerung der Industriewaschmaschine ist zum Beispiel die Anforderung S(3): „*Beim Schleudervorgang darf kein Wasser in der Trommel vorhanden sein*“, sicherheitskritisch und muß in das Sicherheitsmodell aufgenommen werden

Die Sicherheitsanforderungen sagen über die algorithmische Funktionsweise der Steuerung nichts aus. Die Informationen, *wie* eine Waschmaschine sich während des Betriebs verhält, z.B. daß das auf einer Chipkarte gespeicherte Waschprogramm zuerst eingelesen werden muß, bevor es interpretiert werden kann, wird erst in einer *Leistungsspezifikation* modelliert.

Die Aufstellung einer Leistungsspezifikation ist der zweite Schritt bei der formalen Modellierung. Die Leistungsspezifikation definiert das Systemverhalten. Um zu zeigen, daß alle im Sicherheitsmodell definierten Eigenschaften von der Leistungsspezifikation erfüllt werden, muß ein mathematischer Korrektheitsbeweis geführt werden. Dafür bietet das Verifikationssystem von VSE eine umfangreiche Unterstützung. Sie reicht bis zur vollautomatisierten Durchführung von Korrektheitsbeweisen.

Um Programmcode generieren zu können, muß die Leistungsspezifikation noch weiter verfeinert werden. In VSE wird dies mit Hilfe von *abstrakten Programmen* bewerkstelligt. Der Nachweis ihrer Korrektheit erfolgt dadurch, daß ein Nachweis gegenüber der Leistungsspezifikation geführt wird. Da die Leistungsspezifikation ihrerseits schon das Sicherheitsmodell erfüllt, sind dann auch die abstrakten Programme bezüglich des Sicherheitsmodells korrekt.

3.3. Spezifikation der Industriewaschmaschine mit VSE

3.3.1. Modellierungsausschnitt

Die Modellierung im Rahmen der formalen Spezifikation der Industriewaschmaschine beschränkt sich auf die Steuerungssoftware, wobei

sicherheitsrelevante Funktionen, die bisher in Hardware realisiert waren, hauptsächlich in die Software mit aufgenommen wurden. Die Software für den Chipkartenleser und das Bedienpanel wird nicht modelliert, da sie keine Sicherheitsverantwortung besitzt.

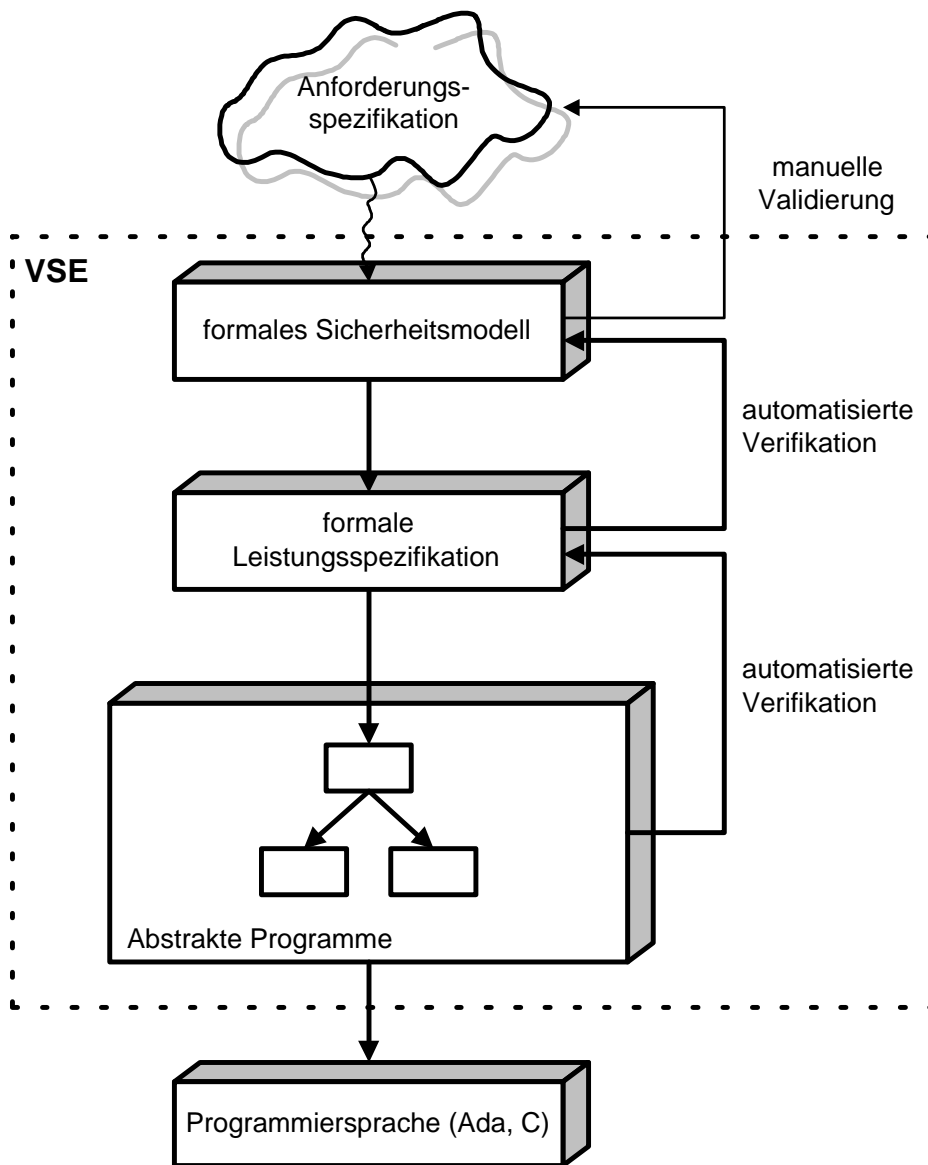


Abbildung 3: Entwicklungsmethodik in VSE

Die bei der Modellierung berücksichtigte Architektur ist in Abbildung 4 zu sehen. Sie umfaßt die Erfassung der Sensorwerte als Eingangsgrößen und liefert Ansteuerungssignale für die verschiedenen Aktoren.

3.3.2. Sicherheitsmodell

Bei der Aufstellung des Sicherheitsmodells wurden die beschriebenen textuellen Anforderungsspezifikationen analysiert und die sicherheitsrelevanten Anforderungen extrahiert.

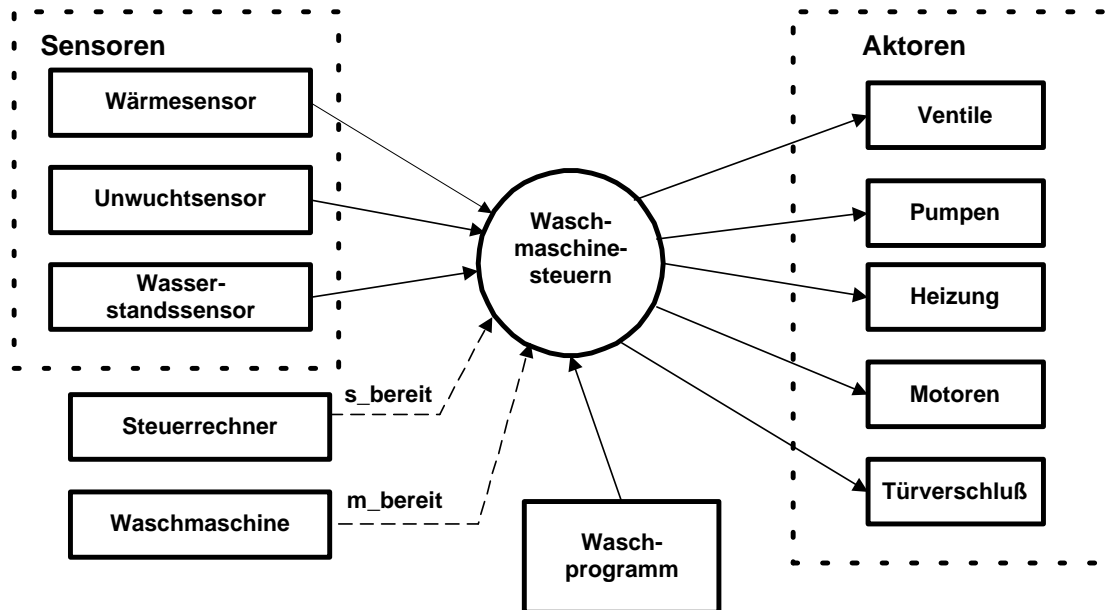


Abbildung 4: Kontextdiagramm der Waschmaschinensteuerung

Das Sicherheitsmodell für die Industriegewaschmaschine ist komplex und umfangreich (ca. 450 Zeilen). Aus diesem Grund wird hier nur beispielhaft die Umsetzung der Anforderung S(3): „Beim Schleudervorgang darf kein Wasser in der Trommel vorhanden sein“ gezeigt. Die Formulierung in der Spezifikations-sprache von VSE sieht folgendermaßen aus:

```
PROC schleudern(Hoehe: Twasserstand; ...)  
    REQUIRES  Hoehe = Standlzu0; /* kein Wasser in der Trommel  
*/  
    ENSURES  Schleudermotor = Schleudern500Umin OR  
            Schleudern1000Umin;
```

Die Anforderung S(3) wird hier direkt in eine Vorbedingung für die Operation *schleudern* umgesetzt. Andererseits wird nach der Ausführung von *schleudern*

eine Zusicherung gegeben, daß der Schleudermotor sich entweder mit 500 oder 1000 Umdrehungen pro Minute dreht.

3.3.3. Leistungsspezifikation

Das Aufstellen der Leistungsspezifikation war für die Steuerung der Waschmaschine problematisch. Die real existierende Automatisierungssoftware setzt auf ein einfaches kooperatives Echtzeitbetriebssystem auf und wurde durch mehrere Tasks implementiert. In der VSE-Version 1.2 können zur Zeit keine Echtzeiteigenschaften modelliert werden. Deswegen konnte obiger Lösungsansatz nicht umgesetzt werden.

Stattdessen wurde in einem zweiten Ansatz versucht, das System durch einen Zustandsgraphen zu modellieren, wobei alle Sensor- und Aktorwerte in einem Tupel zusammengefaßt wurden. Dadurch entstand eine sequentielle Steuerungsoperation, die über eine Endlosschleife aufgerufen werden konnte. Auch dieser Ansatz scheiterte, da die Anzahl der Sensor- und Aktorwerte bei der Spezifikation einen nicht mehr zu beherrschenden Umfang ergeben hätte.

Zuletzt wurden alle über die Steuerungsfunktionen bekannten Informationen zusammengetragen und in ein sequentielles Zustandsübergangsdiagramm (24 Zustände und ca. 50 Zustandsübergänge) umgesetzt. Dieser Weg schien zunächst bei der Aufstellung der Leistungsspezifikation zufriedenstellend. Die Komplexität des Modells am Ende war jedoch sehr hoch und die Spezifikation selbst ist trotz guter Kommentierung schwer lesbar, da die Umsetzung des Zustandsübergangsdiagramms in VSE nur auf einer Ebene erfolgen kann. Dies ist für komplexere Systeme nicht ausreichend. Der Entwickler kann hier schnell dazu verführt werden, viele Vereinfachungen vorzunehmen und so die Realität nur eingeschränkt wiederzugeben, was in vielen Fällen nicht akzeptabel ist.

3.3.4. Abstrakte Programmierung und Korrektheitsbeweise

Die Verfeinerung der Leistungsspezifikation wurde aus Zeitgründen nur ausschnittsweise vorgenommen. Eine Verifikation wurde nicht durchgeführt.

Diese Einschränkungen waren aber für die Ergebnisse nicht ausschlaggebend, da das Ziel die Untersuchung der prinzipiellen Eignung von VSE zur Spezifikation von komplexeren Automatisierungssystemen war.

3.4. Erfahrungen

Die Durchführung einer formalen Spezifikation von Software mit VSE ist für einen Ingenieur aufwendig. Die Erfahrung zeigt, daß zum Lernen und Verstehen der Methode mindestens 3 Monate intensiver Einarbeitung benötigt werden. Um reale Projekte durchführen zu können, muß der unerfahrene Entwickler vorher ein größeres System weitgehend selbständig spezifiziert haben. Dadurch beträgt die Einarbeitungszeit mindestens 6 Monate.

Die Berücksichtigung bestehender Architekturen bei der formalen Spezifikation mit der aktuellen VSE-Version ist, insbesondere im Bereich der Automatisierungstechnik, für komplexe Systeme nur bedingt möglich. Die VSE-Methodik schränkt die Ausdrucksmöglichkeiten und die daraus resultierenden Systemarchitekturen stark ein. Für sequentielle Systeme, deren Architektur keine zusätzlichen Komponenten wie Echtzeitbetriebssystem oder Kommunikationssystem erfordert, ist eine sinnvolle Anwendung der VSE-Methodik schon jetzt durchaus möglich.

In der aktuellen Version unterstützt VSE die Modellierung von Echtzeiteigenschaften nicht. Dadurch wird das Einsatzfeld innerhalb der Automatisierungstechnik stark eingeschränkt. Eine neue, sich in Entwicklung befindende Version, wird bald Abhilfe schaffen.

Die Modellierung komplexerer zustandsbasierter Systeme, wie der Waschmaschine, ist wegen fehlender Hierarchisierungsmöglichkeiten von Zustandsdiagrammen nur auf sehr abstraktem Niveau möglich. Dies ist für die Praxis oft nicht ausreichend.

Bei einer Bewertung der Erfahrungen entsteht der Eindruck, daß die VSE-Spezifikationssprache vorwiegend für den Einsatz für die Spezifikation und

Verifikation von Algorithmen, Datenstrukturen oder sequentiellen Systemen entwickelt wurde. Eine Stärke der Methode ist jedoch die Möglichkeit, ein System in einen sicherheitskritischen und nichtsicherheitskritischen Teil zu spalten und diese Teile mit unterschiedlichen Methoden zu entwickeln [2]. Ein Problem hierbei könnte sein, daß viele Systeme in dieser Weise nicht sinnvoll aufgeteilt werden können bzw. nach einer Aufteilung der sicherheitskritische Teil immer noch sehr komplex ist.

Der Einsatz formaler Methoden hat aber einige entscheidende Vorteile. Durch die Anwendung der VSE-Methodik wird der Entwickler gezwungen, sich mit dem Problem sehr gründlich auseinanderzusetzen. Hierbei ist die Wahrscheinlichkeit, daß Fehler, Inkonsistenzen oder die Unvollständigkeit des Modells während der Spezifikation, und insbesondere während der Verifikation entdeckt werden, sehr hoch. Man spricht deshalb auch nicht nur über die Verbesserung der Qualität, sondern über die Steigerung des *Vertrauens* in das entwickelte System.

Eine formale Entwicklung sollte nicht „breit“ und als alleinige Methodik zur Entwicklung von Softwaresystemen eingesetzt werden. Sie sollte vielmehr als notwendige Ergänzung bei der Entwicklung qualitativ hochwertiger Systeme angesehen werden. Der effiziente Einsatz streng formaler Methoden und Werkzeuge wie VSE ist heute durch Bildung gemischter Entwicklerteams möglich. Sie können z.B. aus Mathematikern/Informatikern und applikationserfahrenen Ingenieuren bestehen.

4. Zusammenfassung und Ausblick

Erfahrungen und Entwicklungen der letzten Jahre zeigen: Softwaresysteme mit hohem Sicherheits- und Qualitätsniveau können nur entwickelt werden, wenn das gesamte System (Hardware, Software und alle anderen Komponenten) einer gleichartigen Sicherheits- und Qualitätspolitik unterliegen. Werden zusätzlich gesetzliche Vorschriften, internationale Normen und Empfehlungen von Verbänden als Maßstab genommen, bedeutet das für die Softwareentwicklung in der nahen Zukunft auch im Ingenieurbereich den Einsatz formaler Methoden. Die nächste Version von VSE wird auch die Spezifikation und Verifikation von

Echtzeitsystemen unterstützen und somit für eine größere Klasse von Automatisierungssystemen anwendbar sein. Werkzeuge wie VSE bereiten den Weg für den Einsatz formaler Methoden und es ist mit *Sicherheit* sinnvoll, sich mit Ihnen zu beschäftigen.

Literatur

- [1] Baur, P. et al: „*The Verification Support Environment VSE*“
IFAC Symposium on Safety, Security and Reliability of Computers, 1992
- [2] Göhner, P.: „*Spezifikation und Verifikation von sicheren Softwaresystemen*“
atp, 4/1995, pp. 24-31.
- [3] Bundesamt für Sicherheit in der Informationstechnik (BSI):
„*VSE Handbuch*“
März 1995
- [4] Canver, E., Gayen, J.-T. und Moik, A.: „*Formale Entwicklung der Steuerungssoftware für eine elektrisch ortsbediente Weiche mit VSE*“
Ulmer Informatik-Berichte, Nr.96-01, Universität Ulm, Februar 1996