

Bau von sicherheitskritischen embedded Systemen mit Hilfe von Datenmodellen

Robert Schachner, RST Industrie Automation GmbH

Peter Schuller, MicroSys Electronics GmbH

Motivation

Es wird heute als selbstverständlich vorausgesetzt, dass technische Geräte, Anlagen, Fahrzeuge Transportmittel und Maschinen immer zuverlässig und sicher funktionieren. Damit diese Selbstverständlichkeit auch immer gewährleistet ist, sind vor allem in Zusammenhang mit elektronischen Steuerungen und Embedded-Systemen, erhebliche Aufwände seitens der Hersteller zu tätigen. Zum einen die zunehmende Komplexität elektronischer Geräte in ihren Hard- und Softwarefunktionen und zum anderen Auflagen seitens Behörden, Sicherheitsstandards für die verschiedensten Marktsegmente und Anwendungskategorien, sowie Normen und Standards, die zu erfüllen sind. Diese Gegebenheiten müssen bereits bei der Konzeption eines elektronischen Produktes berücksichtigt werden und wirken sich maßgeblich auf die Entwicklung sowie den gesamten Produktlebenszyklus aus. Verbunden damit sind natürlich auch verschiedene Vorgehensweisen in der Entwicklung für verschieden Märkte und Anwendungsprofile mit entsprechenden Auswirkungen auf Entwicklungskosten und –zeit.

Wir wollen uns in diesem Artikel nicht um die spezifischen Anforderungen von sicherheitsgerichteten Entwicklungen für Märkte wie die Industrieautomation, der Avionik oder Medizintechnik widmen. Hier sind die Hersteller in der Regel mit den individuellen Gegebenheiten vertraut.

Was wir beschreiben, sind neue Methoden in der Entwicklung von sicherheitsgerichteten Embedded-Geräten, vor allem in Bezug auf die Software-Technik und -Auslegung. Das sind Vorgehensweisen, die bereits vor der Spezialisierung auf die Sicherheitsanforderung individueller Marktanforderungen greifen. Mit einem sog. modellorientierten Ansatz, der auf eine Abstraktion in einem Datenmodell zurückgreift, werden Systemkomponenten modularisiert beschrieben, die nachfolgende, sicherheitsgerichtete Auslegungen vereinfachen. Gleichzeitig ist auch die Adaption eines Designs für Sicherheitsanforderungen in unterschiedlichen Marktsegmente einfacher umzusetzen.

Anhand einiger umgesetzter Beispiele erklären wir, wie Datenmodelle eine sicherheitsgerichtete Entwicklung unterstützen. Dieser Leitfaden ist dabei relativ einfach auch auf andere sicherheitskritische Entwicklungen in der Automation übertragbar.

Einleitung

Früher wurden sicherheitsrelevante Funktionen in Maschinen meist mechanisch oder elektrisch realisiert. Bei einer Stanzmaschine beispielsweise verhindern zwei getrennte Taster, die mit

+ Gebündelte Kompetenz + Offene Technologien + Verbundlösungen aus einer Hand + Kurze Entwicklungszeiten +

beiden Händen gleichzeitig betätigt werden müssen, eine Gefährdung des Bedienpersonals. Auch in komplexeren Anlagen finden sich nach wie vor elektrische Not- Aus-Schalter.

Mit zunehmender Funktionalität jedoch, ermöglicht vor allem durch die elektronischen Komponenten, muss eine industrielle Anlage inzwischen eigene funktionale Sicherheit aufweisen. Sie muss selbstständig in der Lage sein Aktionen durchzuführen, die notwendig sind, um einen sicheren Zustand zu erreichen oder diesen sicheren Zustand aufrecht zu erhalten.

Bei einem medizinischen Tumor-Bestrahlungsgerät zum Beispiel, werden in sehr kurzen Zeitabständen die Strahlungs dosis und die Strahlposition überwacht. Wird eine tolerierte Abweichung überschritten, so muss die Einrichtung unverzüglich abgeschaltet werden, um eine Schädigung des menschlichen Gewebes außerhalb des Tumors unbedingt zu verhindern. In dieser Anlage wird eine Protonenstrahlung mit schweren Magneten zehntel Millimeter genau am Patienten fokussiert. Die Steuerung muss höchsten Anforderungen entsprechen, um die medizintechnischen Auflagen zu erfüllen, redundant ausgelegt sein, um Ausfallsicherheit zu gewähren und hohe Echtzeitvorgaben zu erfüllen. Hierzu wurde eine redundante Embedded-Echtzeitplattform, mit dem Microware OS-9 Echtzeitbetriebssystem, I/O-Komponenten, Kommunikationsinfrastruktur mit Steueralgorithmik entwickelt und zertifiziert. Als verbindendes Element zwischen den Hardwarekomponenten, dem Betriebssystem und der Anwendung fungiert die Middleware „GAMMA“, die als zentrales Element mit einem sog. datenzentrischen Ansatz oder auch Datenmodell, die Realisierung der gesamten Anlage wesentlich vereinfachte.

Grundlagen einer modellorientierten Entwicklung

Modellgetriebene Softwareentwicklung, eine kurze Einführung

Im Wesentlichen führen zwei Aspekte zu diesem Ansatz. Zum einen ist es die zunehmende Komplexität heutiger und künftiger embedded Systemlösungen, die strukturiert und übersichtlich beherrscht werden soll. Das vermeidet die bekannten Datengräber und führt beim Kunden zur Reduktion auf die Kernkompetenz seiner Entwicklungsarbeit. Zum anderen möchte man das Rad nicht zweimal erfinden und Systemfunktionen immer wieder neu programmieren. Vorhandene und künftige Systembausteine sollen sozusagen in Komponentenform, baukastenähnlich einsetzbar und wiederverwertbar sein. Damit einhergehend ist es zwingend notwendig, aufwendige Programmierung einfacher zu gestalten, um die Komplexität zu beherrschen und zusätzlich die Entwicklungszeiten für die Projekte im Zaum zu halten. Bei der modellbasierten Softwareentwicklung versucht man idealerweise aus formalen Modellen automatisch Software zu erzeugen, d.h. es müssen Methoden und Werkzeuge existieren, die es erlauben, einen gewünschten Sachverhalt auf einer abstrakten Ebene zu beschreiben. Die Umsetzung in ausführbaren Code wird dann automatisiert. Beispiele dafür sind Produkte wie Matlab-Simulink von der Firma MathWorks oder Rhapsodie von Telelogic, inzwischen IBM.

Modellierung durch eine datenzentrisches Beschreibungsmodell

- „GAMMA“, ein sog. Softwarestecker als „Middleware“-Komponente -

Ein Softwarestecker (siehe Abb. 1) greift die Erfolgsgeschichte und Idee standardisierter Hardwareinterfaces (z.B. Bussysteme wie VME oder CompactPCI) auf und setzt sie auf die jeweilige Software- und Systemumgebung um. Der neue Ansatz, auf Software-Ebene die Systemkomponenten sozusagen steck- bzw. konfigurierbar auszulegen, enthebt den Entwicklungsprozess von aufwendigen Betriebssystemadaptionen, Treiberanpassungen für Peripherie, I/O-Funktionen und der Integration von Netzwerkfunktionen auf der Basis zeitraubender Programmierung. Das spart über das gesamte Projekt Mannjahre an Entwicklungszeiten ein.

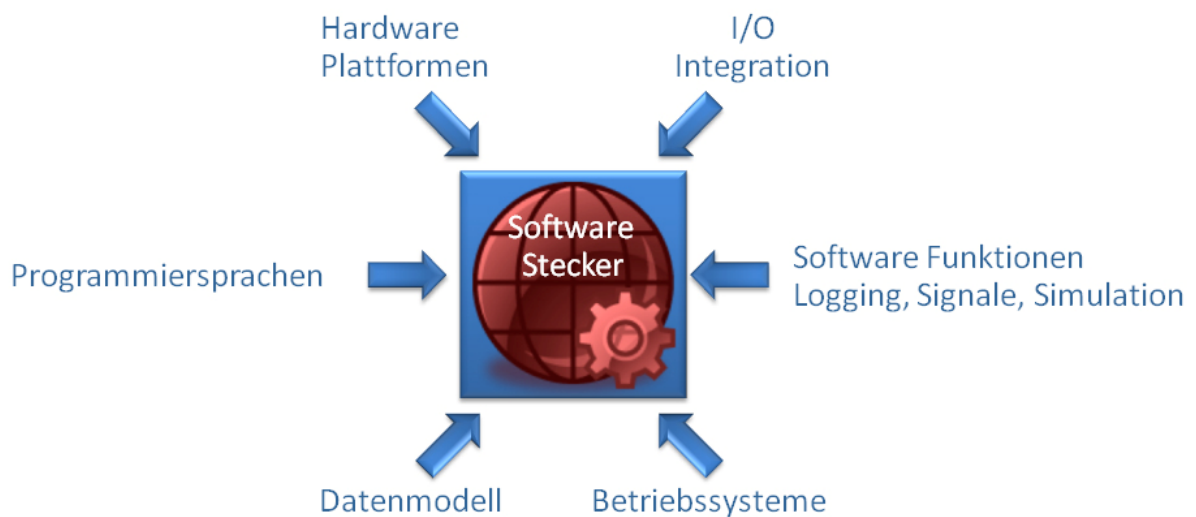


Abb. 1) Der Softwarestecker und die Systemkomponenten, die er verbindet

Der Softwarestecker abstrahiert, konfiguriert und integriert:

- komplette Hardwareplattformen, inkl. CPUs, BUS-, Speicher- und lokales I/O-System
- Betriebssysteme wie Windows, Linux, Microware OS-9 und VXWorks
- Kommunikations- und Netzwerkinfrastrukturen: TCP/IP, Feldbusse wie CAN, EtherCAT, Profinet, etc
- Programmiersprachen
- Grafikserver, wie zum Beispiel XiBase9
- Grafische Entwicklungsumgebungen, z.B. Matlab/Simulink
- Heterogene Rechnerarchitekturen, auch in verschiedenen Netzwerken (meist TCP/IP) integriert
- SPS-Systeme
- Firmenspezifisches Software-Know-how in Form von Komponenten
- über OPC (OLE for process control)-Schnittstellen weitere Techniken, wie z. B. Prozessvisualisierungen

+ Gebündelte Kompetenz + Offene Technologien + Verbundlösungen aus einer Hand + Kurze Entwicklungszeiten +

Ein wesentlicher Bestandteil dieses Softwaresteckers ist eine datenzentrische Organisation der Systemvariablen durch ihre Abbildung in einem Datenmodell

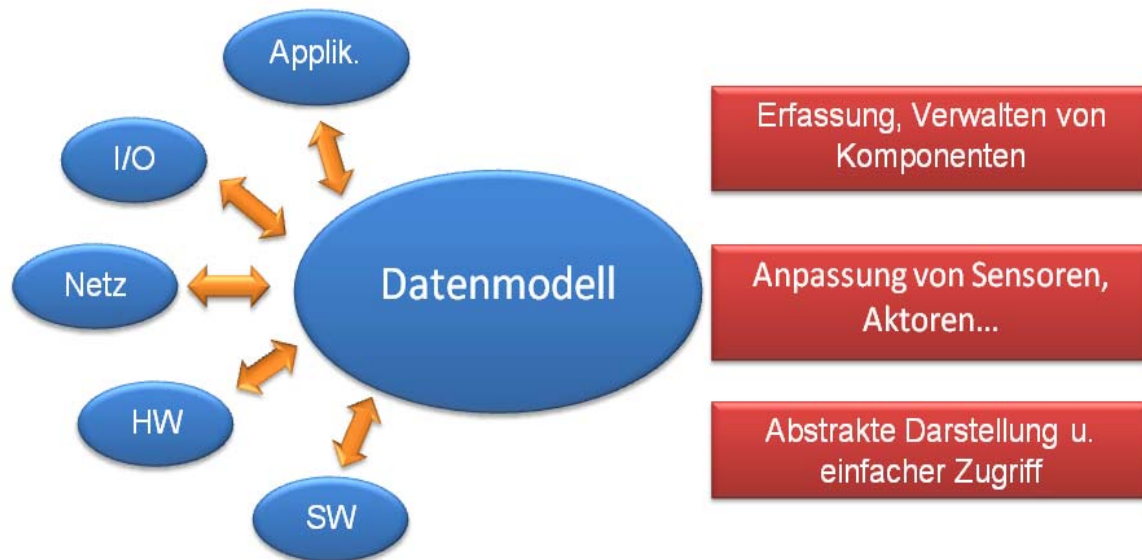


Abb. 2) Das Datenmodell, für die Beschreibung der Systemkomponenten

In diesem datenzentrischen Modell wird die Applikation um ein Softwaremodell gestaltet, bei dem die Daten im zentralen Fokus und vor allem außerhalb der Applikation als eigenes Konstrukt stehen. Die Zugriffe auf diese Daten sind für den Anwender völlig einheitlich, obwohl sie lokal, über Netzwerke, über verschiedene Rechner mit verschiedenen Betriebssystemen und Programmiersprachen hinweg funktionieren.

Die von der Hardware vorgegebenen I/O Funktionen können soweit die Software dafür bereits vorhanden ist, in einem Konfigurationsprozess Teilmodellen zugeordnet werden. Neue Anforderungen werden einfach als Source-Code in die Entwicklungsumgebung integriert. Jetzt können die Funktionen einer gesamten Steuerung wahlfrei mit unterschiedlichen Programmierwerkzeugen erstellt und über das Datenmodell miteinander verbunden werden. Diese können dann unabhängig vom Werkzeug miteinander kommunizieren, sich synchronisieren und die Hardware ansprechen. Simulations- und Logging-Funktionen bringt das Datenmodell für den nachfolgenden Test bereits mit. Auf diese Weise sind Anlagen realisierbar, in denen heterogene Rechnerumgebungen, Daten-Logger, I/O-Systeme, Sensorik und Aktoren transparent miteinander kommunizieren.

Zertifizierung, Zertifizierbarkeit und Entwicklungsprozesse

Um sicherheitskritische Auflagen von Anlagen zu erfüllen, werden Hersteller von embedded Systemen immer mehr in die Pflicht genommen, ihre Produkte gemäß vielfältiger Sicherheitsnormen, -Standards und -Vorgaben anzubieten. Obwohl in der Regel sicherheitsgerichtete Systeme als Einheit getestet und zertifiziert werden, werden auch zunehmend zertifizierte oder zertifizierbare Subkomponenten gefordert, ob notwendig oder

+ Gebündelte Kompetenz + Offene Technologien + Verbundlösungen aus einer Hand + Kurze Entwicklungszeiten +

nicht. Beide Faktoren haben maßgebliche Auswirkungen auf die Produkte, Entwicklungsprozesse und Kosten. Als Hersteller von elektronischen embedded Subsystemen kennt man in vielen Fällen den endgültigen Einsatzzweck des Geräts nicht. Dennoch wäre es wünschenswert bewährte Prozesse aus aufwendigen sicherheitsgerichteten Anwendungen, wie z. B. der Avionikentwicklung sinnvoll, einfach und kostengerecht auf andere Marktsegmente zu übertragen.

Mit der Einführung einer modellorientierten und datenzentrierten Softwareinfrastruktur, wie sie oben vorgestellt wurde, bieten sich viele Vorteilen bereits in einer frühen Phase der Entwicklung auf Komponentenebene eines Geräts; die eine sicherheitsgerichtet Auslegung und spätere Zertifizierung wesentlich erleichtern.

Eigenschaften der Middleware „GAMMA“ hinsichtlich einer sicheren Geräteauslegung

Die Middleware stellt aktive und passiv ausführbare Zugriffsfunktionen zur Verfügung. Sie abstrahiert und strukturiert im Kern die gesamte Hardware inklusive I/O-Funktionen, benötigte Systemkomponenten (Hard- als auch Software) bis hin zu Applikationen.

Wesentliche Merkmale, die Sicherheitsaspekte adressieren sind:

- Die verwendeten Funktionen aus der Middleware sind bereits sehr stabil und haben meist bereits vorgefertigte Testtreiber, die in dem Systemtest mit einbezogen werden können.
- Gerade systemübergreifende Kommunikationsstrukturen, wie zum Beispiel Prozesse mit Prozessen oder Threads mit Threads, sind bereits vorhandene Grundfunktionen auf dem System. Im Netzwerk mit den entsprechenden Testtreibern erleichtern sie die Entwicklung immens.
- Die individuelle Programmierung beschränkt sich auf die reine Applikation und deren Test (bei Unit Tests)
- Code-Reviews werden einfacher und transparenter, da die Applikationslogik wesentlich kompakter getrennt von den unterlagerten Hard- und Softwareschichten ist.
- Die Ausprägung sicherheitsgerichteter Systeme als redundante Einheiten wird von der Middleware vollständig unterstützt.
- Bereits integrierte Testfunktionen, wie Simulation und Daten-Logging erleichtern den Aufbau von Testszenarien. Darauf aufbauend gibt es am Markt bereits professionelle Testumgebungen, die den gesamten Entwicklungsprozess unterstützen.
- Aus Sicherheitsgründen können die eingebauten Testfunktionen für das endgültige Testsystem abgeschaltet werden. Damit kann mutwillige Manipulation des Systems ausgeschlossen werden.

Mit diesem Handwerkszeug bestehen Strukturen, die in komplexen Abhängigkeiten die Übersicht organisieren und erleichtern. Vom Design zur Prototypenentwicklung, über die Serieneinführung und über den gesamten Produktlebenszyklus bleiben diese Strukturen erhalten und unterstützen Anpassungen und Modifikationen. Für Tests, Validierungen und Zertifizierungen ist dieses zentrale

+ Gebündelte Kompetenz + Offene Technologien + Verbundlösungen aus einer Hand + Kurze Entwicklungszeiten +

„Tool“ natürlich hinsichtlich der Vereinfachung, Beherrschung, Übersichtlichkeit und Organisation von komplexen Abläufen ein wesentliche Hilfe.

Sehen wir uns in den nächsten Abschnitten an verschiedenen umgesetzten Lösungen exemplarisch an, in dem „GAMMA“ eine wichtige Rolle einnimmt, ein sicherheitsgerichtetes Design umzusetzen.

Die Wahl der richtigen Hardware

Die Anforderungen an die funktionale Sicherheit einer Anlage hat Auswirkungen auf die Wahl der Systemkomponenten. Mit steigenden Sicherheitsanforderungen müssen entsprechende Maßnahmen getroffen werden, z. B.:

- Hardware mit erweiterten Temperaturbereichen
- Hardware mit eingebauten Sicherheitsfunktionen zur Selbstüberwachung, wie zum Beispiel einer „Watchdog“-Funktion.
- Überwachungsfunktionen mit programmierbarer Logik, die auch geeignet für Zertifizierungen sind.
- Möglicherweise redundante Strukturen, d.h. Mehrfachauslegung wichtiger Systemeinheiten, wie z. B. CPU-Boards, die sich gegenseitig überwachen und bei Ausfall entsprechend die aktive Rolle übernehmen

In der erwähnten Bestrahlungseinrichtung, wurde eine Kombination aus den oben beschriebenen Maßnahmen gewählt:

- Wichtige Sensoren sind redundant ausgelegt und Daten werden mehrfach erfasst.
- Schaltfunktionen werden redundant ausgeführt.
- Die zentrale Rechereinheit wird durch ein parallel arbeitendes CPU-Board überwacht, das im Problemfall die nötigen Sicherheitsfunktionen ausführt.

Gut geeignet als Hardware-Plattformen sind solche mit Bussystemen, die den Betrieb mehrfach vorhandener Prozessoreinheiten ermöglichen. Klassisch wäre hier ein VME-Bus-System zu nennen, das über einen passiven Backplane Bus miteinander verbunden ist (Abb. 3) und im Bestrahlungssystem eingesetzt wird. Ein alternativer und moderner Ansatz ist eine microTCA-Architektur, wie in Abb. 4 dargestellt. Es erlaubt komplett getrennte Systeme in einem Rack zu integrieren, die über eine PCI-Express oder Ethernet-Verbindung miteinander kommunizieren.



Abb. 3) Redundantes VME-Bus-System



Abb. 4) Redundantes microTCA-System

Doch auch einfachere Konfigurationen sind denkbar. Redundant vorhandene embedded Boards, deren Überwachung über eine Kommunikationsschnittstelle wie Ethernet verfügt, sind grundsätzlich für sicherheitsgerichtete Hardwareauslegungen geeignet.

Die Wahl des richtigen Betriebssystems

Betriebssysteme organisieren die Hardware, I/O-Funktionen, Netzwerke und sind die Schnittstelle zu den Anwendungsprogrammen. Mit zunehmender Aufgabenvielfalt und Anforderungsprofilen steigt deren Komplexität. Ihre Flexibilität ermöglicht vielfältigste Einsatzzwecke, d.h. eine eindeutige Ausprägung für alle Anforderungen gibt es nicht. Vom Betriebssystemhersteller kann deshalb auch nicht erwartet werden, sicherheitsgeprüfte Versionen für alle Einsatzzwecke sozusagen vorrätig zu haben. Es gibt zertifizierbare Betriebssysteme, z.B. für die Luftfahrt, deren Einsatz aber ökonomisch in der Automation nicht sinnvoll ist. Hauptsächlich werden dabei Vorgaben für stringent einzuhaltende Entwicklungsprozesse und entsprechende Dokumentation bereitgestellt, die den Zertifizierungsaufwand reduzieren. Das bedeutet jedoch hohe Anschaffungs- und Entwicklungskosten und ggf. lange Testzeiten im Projekt

In der Automation empfiehlt es sich deshalb auf andere Aspekte zu achten, wie zum Beispiel

- Existiert das Betriebssystem ausreichend lange am Markt für den beabsichtigten Einsatzzweck, d.h. hat es sich bewährt, in Funktion, in vergleichbaren Applikationen und durch die Unterstützung seitens des Herstellers?
- Ist es modular aufgebaut, d.h. können Funktionen spezifisch für die Applikation ausgewählt, und konfiguriert werden, um unnötigen Ballast auszugrenzen und um die Übersichtlichkeit zu behalten?
- Sind Sicherheitsmechanismen, wie automatische „Cyclic Redundancy-Checks“ integriert, die auf Modulebene Codekonsistenz sicherstellen?
- Unterstützt es „Memory Managements Units“ für die sichere Speicherorganisation?
- Gibt es Einsicht oder Zugriff auf den Sourcecode, falls eine Zulassungsbehörde dies fordert?
- Gibt es Unterstützung seitens des Herstellers für Zertifizierungen, falls notwendig?

In welcher Form das Betriebssystem für den jeweiligen Anwendungsfall Normen erfüllen oder gesondert zertifiziert werden muss, helfen entsprechende Einrichtungen, wie z.B. der TÜV zu klären. Sind die oben aufgeführten Anforderungen erfüllt, ist jedoch bereits eine gute Basis für den Bau eines sicheren Systems gegeben.

Sicherheitsgerichtetes Design und die Aufgabe der „Middleware“ GAMMA

Ein Weg wie fehlertolerante Systeme „gebaut“ werden können, ist wie oben bei der Hardwareauswahl bereits erwähnt, eine redundante Auslegung wichtiger Komponenten. Das können sein:

- Hardware (strukturelle Redundanz)
- Information
- Zeit
- Software (funktionelle Redundanz)

Je nach Sicherheitsanforderungen wird für das Systemdesign in der Risikoanalyse ein Mix aus Redundanzen zugrunde gelegt. Um sicherheitskritische Anforderungen nachzuweisen, wird natürlich sehr viel Aufwand in die Tests, Validierung der Ergebnisse und ggf. für die Zertifizierung zu legen sein. Man geht heute davon aus, dass 70 % des gesamten Entwicklungsaufwands eines sicherheitskritischen Geräts in die Verifizierung und Validierung fließen. Nachdem „GAMMA“ die grundsätzlichen Kommunikations- und Testmethoden für redundante Systeme bereits beinhaltet, werden diese Entwicklungsschritte wesentlich vereinfacht, verkürzt und übersichtlicher. Der Einsatz von formalen Methoden, worunter auch die Verwendung von „GAMMA“ gerechnet werden kann, ist deshalb für höhere Sicherheitsintegritäten zwingend notwendig.

Ansprechen von Prozessvariablen in redundanten Systemen mit „GAMMA“

Die erwähnten Redundanzmodelle lassen sich relativ einfach über gegenseitige Zugriffsmethoden realisieren. Wie bereits dargestellt, abstrahiert GAMMA Systemkomponenten in einer virtuellen Datenstruktur. Sie ist aufgebaut wie ein Filesystem. Prozessvariable, wie Sensordaten, Stell- und Regelgrößen werden über sog. „Select Strings“, die wie Filepfade aufgebaut sind, angesprochen und zwar einheitlich und über die Systemkomponentengrenzen hinweg. Redundante Funktionen sind dabei natürlich mit eingeschlossen. Prüf- und Testverfahren, die auf diese Mechanismen zurückgreifen, sind übersichtlich, einfach zu modifizieren, zu automatisieren, nachvollziehbar und dokumentierbar. Aufwendige „Low level“-Programmierung entfällt dabei.

Sicherheitsgerichtete Applikationsentwicklung mit diesem Ansatz

Mit einer auf die jeweilige Hardware-Architektur konfigurierten „GAMMA“-Middleware, reduziert sich der Aufwand für die Applikationsentwicklung auf die eigentliche Anwendung. Sensorik, andere Computer mit anderen Betriebssystemen, Redundanzen oder Multicore-Anwendungen sind für diese Architektur transparent und müssen in der Programmierung der Anwendung nicht weiter berücksichtigt werden. Die Entwicklung kann sich hier auf die reine Anwendungslogik fokussieren. Dieser Schritt ist natürlich applikations- und marktspezifisch, so auch die damit

+ Gebündelte Kompetenz + Offene Technologien + Verbundlösungen aus einer Hand + Kurze Entwicklungszeiten +

zusammenhängenden Sicherheitsnormen und –richtlinien. Durch diesen Fokus und die vorgegebene unterliegende Datenstruktur verringert sich der Gesamtaufwand der Entwicklung. Im Produktlebenszyklus werden die Wartbarkeit und Technologieanpassungen der Geräte und Anlagen wesentlich vereinfacht.

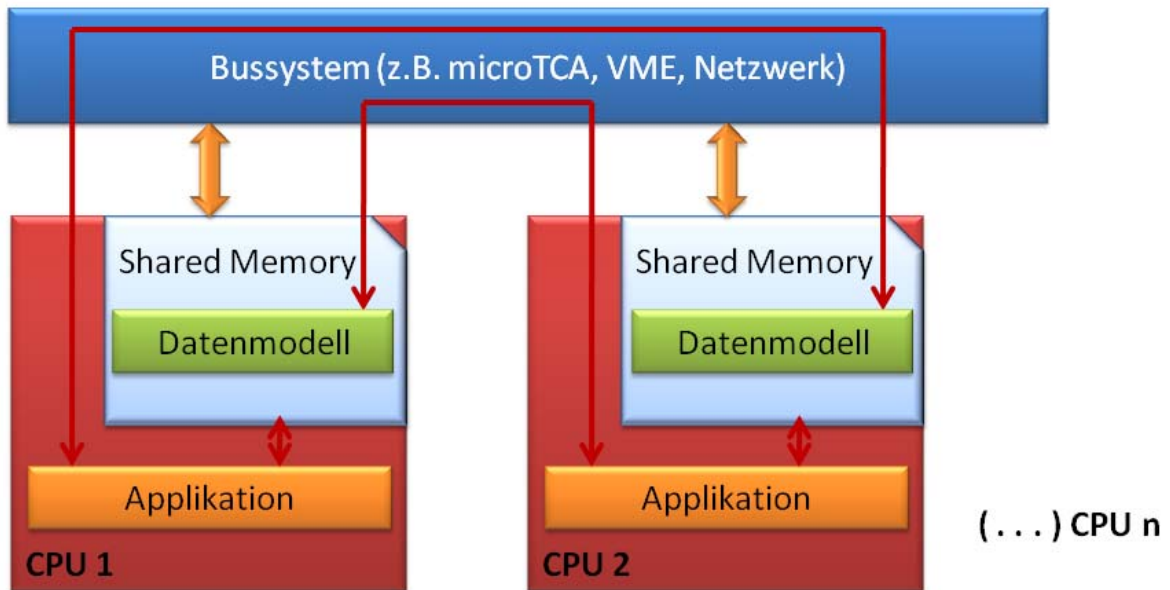


Abb. 5) Hard- und Softwareaufbau eines redundanten Systems mit Datenmodell

Die Problematik der Zertifizierung

Das Ziel einer Zertifizierung?

Zertifizierte Geräte sollen in verschiedenen Industriebereichen Sicherheitsnormen bzw. definierte Standards erfüllen. Damit soll sichergestellt werden, dass ein sicherer und zuverlässiger Betrieb gemäß der Normierung gewährleistet ist. Gleichzeitig sollen damit Haftungsfragen, wie beispielsweise die Produkthaftung geregelt werden.

In sicherheitskritischen Themenfeldern, wie zum Beispiel der Avionik, Automobiltechnik, Medizintechnik und Automation, hat man sich auf gemeinsame Sicherheitsstandards geeinigt. Das sind in erster Linie Prozessnormen, die die notwendigen Schritte im Entwicklungsprozess beschreiben, festlegen und die vor allem auch immer nachvollziehbar sein müssen (Merke: Produkthaftung). Der Hersteller muss in der Entwicklung diesen Vorschriften entsprechen, ist aber in der individuellen Erfüllung derselben wahlfrei, d.h. er kann Methoden seines Know-hows einfließen lassen. Hier können spezielles Know-how und innovative Vorgehensweisen signifikante Wettbewerbsvorteile bedeuten.

Wie wird zertifiziert?

Jeder sicherheitskritische Bereich hat seine eigenen Regeln zusammengefasst in entsprechenden regionalen und überregionalen Normen.

+ Gebündelte Kompetenz + Offene Technologien + Verbundlösungen aus einer Hand + Kurze Entwicklungszeiten +

Beispiele für Normen sind:

- ISO 9001 - Qualitätsmanagement Norm für innerbetriebliche Abläufe
- IEC 61508 - funktionale Sicherheit in elektrischen Systemen
- DO-178 B/C - Software Considerations in Airborne Systems and Equipment Certification
- ISO 13485 - Design und Herstellung von Medizinprodukten

Alle diese Normen gehen prinzipiell auf den Entwicklungsweg nach dem sog. V-Modell zurück. Es beschreibt eine Vorgehensweise für die Planung und erfolgreiche Durchführung von Systementwicklungen. Der Projektablauf soll damit strukturiert und nachvollziehbar ablaufen; die Qualität des Geräts gemäß der Norm erfüllt werden. Von der Anforderung wird stufenweise ein Entwurf/Prototyp erstellt, der dann wiederum stufenweise in die Serienproduktion überführt wird. Werden Fehler festgestellt und behoben, muss dasselbe Verfahren wieder zyklisch durchlaufen und entsprechend dokumentiert werden. Zertifizierungen können manuell mit entsprechenden Formularen durchgeführt werden, die die Prozessstufen vorgeben. Zur Beschleunigung finden aber auch zunehmende rechnergestützte Verfahren Anwendung.



Abb. 6) Entwicklungsschritte nach dem V-Modell

Zertifizierung unterstützt durch „GAMMA“

Für „GAMMA“ gibt es entsprechende Testwerkzeuge, die abstrahieren, strukturieren, die Validierung und Verifikation maßgeblich mitgestaltet und viele Funktionen automatisch dokumentieren. Die Testwerkzeuge beginnen bei einfachen Unit Tests, gehen über zu requirementsbasierten Werkzeugen basierend auf Doors oder zu Testscripts, die automatisch aus Modellen generiert werden (UML); Beispiel für das letztere sind die Werkzeuge ITE (FTI Group) und CETES Kölsch & Altmann).

In „GAMMA“ sind zum Beispiel die wesentlichen Anforderungen an die Tests für die Zertifizierung bereits im Modell enthalten., das sind z. B.:

- Simulation von Prozessabläufen unter Rückgriff auf die entsprechenden Variablen
- Daten -Logging

Mit der Simulation lassen sich alle möglichen und wahrscheinlichen Systemzustände, abgebildet auf Prozessgrößen und I/O-Funktionen simulieren. Prozesszustände innerhalb und außerhalb erlaubter Grenzen sind mit der Applikation validierbar. Über das Daten -Logging wird das entsprechende Systemverhalten aufgezeichnet. Die Aufzeichnungen wiederum stehen für den funktionalen Nachweis der Zertifizierung zur Verfügung. Mit voll automatisierten Testabläufen kann der Testaufwand gerade in der letzten Phase sehr stark reduziert werden. Der Aufwand individuelle Testsoftware dafür zu erstellen wird durch diesen Ansatz vermieden. Die Middleware steht als Grundlage für Test und die Datenaufzeichnung zur Verfügung und bietet darüber hinaus Schnittstellen für die Ankopplung zusätzlicher externer Test- und Prüfverfahren.

Fazit

Die Komplexität von embedded Systeme in der Automatisierung nimmt in allen Bestandteilen zu. Normen und Standards setzen Rahmen auch im Bereich der Sicherheitsanforderungen, die steigende Anforderungen im Entwicklungsprozess erfordern. Entsprechendes Know-how im Umgang mit sicherheitsgerichteten Designprozessen sollte für die vielfältigsten Märkte zur Verfügung stehen. Für die Prozessabläufe in der Entwicklung wird die Unterstützung durch formale Methoden für die Nachvollziehbarkeit der Abläufe, der Verifikation und Validierung eine zwingende Notwendigkeit. Ein bewährtes Werkzeug wie „GAMMA“ kann hier wertvolle Dienste leisten, schneller und kostengünstiger an das Ziel eines erfolgreichen, sicherheitsgerichteten Produkts für die Automation zu kommen.

Prolog

Bündeln von Kompetenzen in der Unternehmensvereinigung Embedded4You e.V

Die Mitglieder von Embedded4You arbeiten teilweise bereits seit Jahrzehnten in gemeinsamen Projekten erfolgreich zusammen und haben dabei technologisch Führendes für ihre Kunden auf die Beine gestellt. Dieser Hintergrund führte zu dem Entschluss, diese Stärken zu bündeln, neue auf ihren Gebieten führende Partner zu gewinnen und diese gemeinsame Kompetenz dem Markt unter dem Dach von Embedded4You anzubieten. Der Kunde erhält dabei aus einer Projektführung, unter Einbeziehung der Lösungen weiterer Partner, das für ihn optimale Produkt. Er kann dabei über lange Projektlaufzeiten flexibel auf die für ihn geeigneten Technologien zugreifen und wird somit unabhängiger in seiner Wahlfreiheit. Es können sich dadurch neue Einsparungspotenziale ergeben und Projekte können schneller und einfacher neuen Techniken folgen.

Im Hinblick auf erfolgreiche Automatisierungslösungen für seine Kunden bündelt die Vereinigung ihre Aktivitäten in den Bereichen:

- Marketing, Vertrieb und Geschäftsentwicklung
- Gemeinsamer Projektarbeit
- Entwicklung neuer gemeinsamer Technologien

Die Anforderungen der heutigen Zeit führen zwangsläufig zu immer spezielleren Werkzeugen. Deshalb wird es immer wichtiger kundenspezifische Lösungen aus entsprechenden Baukästen effektiv zusammenzustellen.

Middleware, spezialisiert auf die embedded Welt wie GAMMA und nicht auf IT Strukturen, stellen einen wichtigen Baustein für diese Entwicklung dar.

Embedded4You mit ihren spezifischen Kompetenzen im Bereich Diagnose- und Test-Tools stehen zusätzlich für den „Feinschliff“ der Applikationen zur Verfügung. Für sicherheitsgerichtete Lösungen ist darüber hinaus das gesamte Know-how-Spektrum von der Avionik, über die Medizintechnik, für automotive Lösungen und bis zu Automationsanwendungen sozusagen „im Haus“.

Innerhalb der Forschungsaktivität „Software Plattform Embedded Systems 2020 – SPES 2020“ (siehe auch: <http://spes2020.informatik.tu-muenchen.de/>), erhält E4You darüber hinaus eine Förderung, dieses Ziel technologisch umzusetzen und in neuen Produkten dem Markt anzubieten.

„Unser Ziel ist es, dem Markt eine offene Plattform für Hard- und Software-Lösungen anzubieten, in der alles einfach, wie in einem Baukastensystem zusammenpasst und auch individuell von unseren Kunden konfiguriert, erweitert und über den gesamten Produktlebenszyklus gewartet werden kann. Wir stellen dem Markt somit einen offenen „Softwarestecker zur Verfügung“, der ähnlich zur Programmierung traditioneller SPS-Lösungen, aber auch die einfache Integration individueller und herstellerunabhängiger embedded Lösungen ermöglicht“, sagt Peter Schuller, 1. Vorstand von E4You e.V.

Aktuelle Mitglieder von Embedded4You (www.embedded4ou.com)

Firma	Email	Kompetenz
aicas GmbH	www.aicas.de	Real Time Java JAMAICA <ul style="list-style-type: none"> ▪ Automation ▪ Luftfahrt & Verteidigung ▪ sicherheitsgerichtet
BEG Bürkle GmbH	www.beg-buerkle.de	Industrie PC <ul style="list-style-type: none"> ▪ standard ▪ kundenspezifisch
coming GmbH	www.coming-it.de	Application Engineering <ul style="list-style-type: none"> ▪ Automation ▪ Automotive
Elma Trenew GmbH	www.elma.de	Rack Systems (rugged) <ul style="list-style-type: none"> ▪ Automation ▪ Luftfahrt & Verteidigung
Euro Systems GmbH	www.euro-systems.de	Engineering, Zertifizierung <ul style="list-style-type: none"> ▪ Luftfahrt & Verteidigung ▪ Automotive ▪ Automation
fortiss GmbH	www.fortiss.org	Wissenschaft und Forschung <ul style="list-style-type: none"> ▪ software-intensive Systeme
FTI Group	www.ftigroup.net	Engineering, Zertifizierung <ul style="list-style-type: none"> ▪ Luftfahrt & Verteidigung ▪ Automotive ▪ Automation
Hochschule München	www.ee.hm.edu	Angewandte Wissenschaften <ul style="list-style-type: none"> ▪ Automation ▪ EtherCAT
IMACS GmbH	www.imacs-gmbh.de	Embedded Lösungen für Steuerungsaufgaben <ul style="list-style-type: none"> ▪ UML-Tool, Soft SPS ▪ kundenspezifische Hardware ▪ Anwendungsentwicklung
Kölsch & Altmann GmbH	www.koelsch-altmann.de	Software Engineering <ul style="list-style-type: none"> ▪ Automation & Verteidigung ▪ Automotive ▪ IT
MicroSys GmbH	www.microsys.de	Embedded System-Lösungen <ul style="list-style-type: none"> ▪ Platforms HW & SW ▪ RTOS, Standard OS ▪ Feldbusse
N.A.T. GmbH	www.nateurope.com	Kommunikation und Automation <ul style="list-style-type: none"> ▪ MicroTCA ▪ EtherCAT ▪ Networks
Protos GmbH	www.protos.de	Model based S/W development <ul style="list-style-type: none"> ▪ domänenspezifische Sprachen und Tools ▪ modellbasierte embedded Applikationen ▪ Codegeneratoren

Firma	Email	Kompetenz
RST Industrie Automation GmbH	www.rst-automation.com	Datenzentrische Modelle, Integration <ul style="list-style-type: none"> ▪ Automation ▪ Luftfahrt ▪ Test, Verifikation
sepp.med GmbH	www.seppmed.de	Qualitätssicherung, Softwareengineering <ul style="list-style-type: none"> ▪ Automation ▪ Medizintechnik ▪ IT
XiSys Software GmbH	www.xisys.de	Embedded Grafik <ul style="list-style-type: none"> ▪ Automation ▪ Medizintechnik

Embedded4You e.V. ist auf der SPS/IPS/Drives vom 23. – 25. November 2010 in Nürnberg und stellt seine Neuerungen dem interessierten Fachpublikum vor.

Die Autoren

Dipl. Ing. (FH) Robert Schachner studierte Datentechnik an der Fachhochschule München. Sein beruflicher Werdegang führte ihn von der Hardwareentwicklung bei Motorola zur Firma RST Industrie Automation GmbH in Ottobrunn. Seit 1993 ist Herr Schachner Geschäftsführer und beschäftigt sich mit Vertrieb und der Umsetzung von Konzepten auf Basis des datenzentrischen Modells GAMMA. In Embedded4You ist er verantwortlich für die Pressearbeit und die Umsetzung gemeinsamer Technologien.



Dipl. Ing. Peter Schuller studierte an der TUM-München Nachrichtentechnik. Anschließend führte ihn seine berufliche Entwicklung über die technische Vertriebsunterstützung zu Funktionen im Vertrieb, Marketing und in der Geschäftsentwicklung in den Unternehmen Intel, Microware, Bsquare, RadiSys und der MicroSys Electronics GmbH in Sauerlach bei München. Im der Vereinigung Embedded4You ist er erster Vorsitzender.

