



Insight

Industry



**Software-Architektur -
der Schlüssel zum Erfolg**

**Embedded Systems -
in allen Branchen zu Hause**

**SysML -
Klarheit bei den Schnittstellen**



VORWORT

Liebe Leserin,
Lieber Leser,

im Hause Berner & Mattner hört und liest man häufig den Ausdruck "Embedded Systems", welcher mit der Integration in den Mutterkonzern Assystem noch einmal an Wichtigkeit hinzugekommen hat. Aus diesem Grund beschäftigt sich diese Ausgabe unseres "Insight Industry" mit dem Begriff "Embedded Systems" und den unterschiedlichen Sichtweisen auf eingebettete Systeme.

Bedingt durch die vielfältigen Einsatzmöglichkeiten und die unterschiedlichsten Anforderungen ist eine allgemeingültige Charakterisierung von Embedded Systems nur schwer möglich. Aus Sicht von Berner & Mattner

steht der Begriff für ein Konzept zur Einbettung von Rechnerleistung in technische Produkte, um sie damit wertvoller und leistungsfähiger zu machen. Man versteht darunter also Computersysteme, die in Geräten, Anlagen und Maschinen eingebettet sind und ganz spezielle Anwendungen "abarbeiten". Sie bilden soft- und hardwaremäßig eine funktionale Einheit, die darauf ausgelegt ist, ausschließlich die definierte Aufgabe zu erfüllen. Neben der Echtzeitfähigkeit dieser Systeme findet sich häufig die Randbedingung, dass der Aufwand für die Softwareentwicklung minimiert wird - und dies alles unter dem Kosten- und Zuverlässigkeitsaspekt.

Typischerweise findet man solche Systeme in der Luft- und Raumfahrttech-

nik, in Eisenbahnsystemen oder in der Automation und Automobiltechnik – also genau den Markt Bereichen, die Berner & Mattner direkt mit seinen Lösungen zur Optimierung von Entwicklungsaufgaben adressiert.

Lassen Sie sich beim Lesen der spannenden Artikel meiner Kollegen und Kolleginnen in die Welt der Steuergeräte "entführen" und entdecken Sie dabei vielleicht die eine oder andere Neuigkeit. Viel Freude bei Ihrer Entdeckungsreise wünscht Ihnen

Ihr

Dr. Christian Hock
Leiter Geschäftsbereich Industry

IMPRESSUM

Herausgeber:
Berner & Mattner Systemtechnik GmbH
Erwin-von-Kreibitz-Str. 3
80807 München
Tel. +49 (0) 89 608090-0
Fax +49 (0) 89 6098182
www.berner-mattner.com
marketing@berner-mattner.com

Redaktion und Gestaltung:
Simone Horra, Dr. Christian Hock
mit Dank an die Autoren der Beiträge

© Berner & Mattner / November 2011

INHALTSVERZEICHNIS

Software-Architektur – Von der impliziten zur expliziten Architektur	3
Elektromobilität – Embedded Software in der Ladeinfrastruktur	6
Motorsteuerungen – Requirements Engineering	7
Embedded Systems in Space – Spin-Ins aus anderen Branchen	8
Bedieninterfaces – Konsistente HMIs für Defence Electronics	9
DB Netz AG – Schnittstellenspezifikationen durch SysML-Modellierung	10
Software-Standards für Satellitenanwendungen – Entwicklungshandbuch	13
embedded world 2012 – B&M präsentiert "Industrial Embedded Systems"	14
Kurznachrichten	14
Stellenanzeigen	15

Software-Architektur für Embedded Systeme

Von der impliziten zur expliziten Architektur

Eine dokumentierte Software-Architektur hat klare Vorteile. Produkte sind anpassungsfähiger und zukunftssicherer. Außerdem erleichtert das Wissen um Schnittstellen, Kommunikationsbeziehungen und Ressourcen einzelner Komponenten deren Anpassung an neue Hardwarebedingungen und Funktionsanforderungen. Im Bereich der Unternehmenssoftware ist das Festlegen einer Software-Architektur heutzutage Standard, doch bei der Embedded Software wird dieser Aspekt häufig noch vernachlässigt.

Nicht nur bei Neuentwicklungen, sondern auch bei Modernisierungen bestehender Embedded-Lösungen lohnt es sich, einen Software-Architekten einzubinden, um Weiterentwicklungen auf ein zukunftssicheres Fundament zu stellen.

Unter Software-Architektur versteht man die strukturierte Anordnung der Komponenten eines Systems mit Informationen zur Kommunikation zwischen den Komponenten sowie deren Abbildung auf Hardware- oder Softwareressourcen. Eine frühzeitige und projektbegleitende Unterstützung durch einen den einzelnen Entwicklerteams möglichst neutral gegenüberstehenden Software-Architekten ist die ideale Lösung. Eine Reihe von Aufgaben kann man dieser Instanz zuordnen, in vielen Bereichen geht es aber auch einfach um Moderation. Eine zentrale Aufgabe der Embedded Software-Architektur ist es, nicht nur die funktionalen Anforderungen an die Software zu berücksichtigen, sondern auch nichtfunktionale Anforderungen zu erfassen, und dies vor dem Hintergrund einer Lifecycle-Prognose. Un-

ter diese nichtfunktionalen Anforderungen fallen so komplexe Punkte wie Wartbarkeit, Modifizierbarkeit, Skalierbarkeit, Portierbarkeit, Konnektivität, Benutzbarkeit, Zuverlässigkeit, Sicherheit, Leistungsfähigkeit, Effizienz oder Testbarkeit.

Begründete Technologieentscheidungen

So vorteilhaft ein eingespieltes Entwicklerteam für die Kommunikation

ist, so sehr kann es auch durch die Erfahrung aus Vorgänger- oder ähnlichen Projekten in seiner Lösungsfindung eingeschränkt sein. Bewährte Lösungsansätze werden dann oft unreflektiert wiederholt. Neue, möglicherweise besser geeignete und insbesondere flexiblere Technologien bleiben somit unberücksichtigt.

Daher kann es zum Beispiel zu den Aufgaben eines Software-Architekturteams oder externen Beraters gehö-



ren, neue Technologien zu evaluieren und auf Eignung für Projekte zu prüfen sowie neue Toolsets zur Verfügung zu stellen. Künftig benötigte Komponenten müssen identifiziert werden. Wenn es darum geht, für einzelne Komponenten optionale Technologien vorzuschlagen, bewährt sich häufig das Einschalten eines neutralen Architekturbeauftragten.

Es geht um die Diskussion von Design-Entscheidungen, die Definition von Schnittstellen und die Entwicklung von Coding Conventions. Unter der Moderation eines Software-Architekten erfolgen diese Entscheidungen bewusster und werden besser dokumentiert. Nur so bleiben die Zusammenhänge innerhalb eines Systems auch für Entwickler nachvollziehbar, die später einmal an ihm arbeiten müssen und nichts von den impliziten Annahmen und teaminternen Gepflogenheiten seiner Schöpfer wissen.

Implizite Architekturen

In der täglichen Praxis der Embedded-Entwicklung sieht es leider meist anders aus. Zeit- und Kostendruck verhindern oft, dass Gedanken und Arbeit in eine systematisierte Software-Architektur fließen. Aber Fakt ist: Jedes Softwaresystem hat eine Architektur, auch wenn diese nicht explizit modelliert wurde. Entstehen diese Strukturen ungeplant aus der Dynamik der Komponentenentwicklung heraus, tut dies der Funktionalität der Software zunächst keinen Abbruch. Probleme treten erst später auf, wenn es um Wartung, Skalierung oder Portierung geht. Im Nachhinein rächt sich dann zum Beispiel die Wahl einer bestimmten Programmiersprache oder einer Schnittstelle, die unter dem Zeit- und Kostendruck im Entwicklungsprozess schnell und dazu noch undokumentiert getroffen wurde. Die Kommunikation zwischen einzelnen, nicht klar getrenn-

ten Komponenten hat man möglicherweise nur oberflächlich gelöst, statt bereits bei der Wahl der Technologien und Sprachen darauf zu achten, Reibungspotenziale zu minimieren. Wird ein Softwareprojekt nicht in Bezug auf seinen gesamten Lebenszyklus gesehen, besteht bei Embedded-Projekten auch die Gefahr, dass ungewollte Abhängigkeiten zur Entwicklungs-Hardware entstehen. Die Anpassung an neue Hardwaregenerationen kann dadurch sehr aufwändig oder gar unmöglich werden.

Nachträgliche Analyse

Eine klar strukturierte und dokumentierte Software-Architektur vermisst man erst dann, wenn es bei der notwendigen Weiterentwicklung Fragen und Schwierigkeiten gibt. Oft sollen wesentliche Teile der Software beibehalten, aber z. B. an ein neues Benutzerinterface angeschlossen werden. Manchmal ist der Controller, für den sie geschrieben wurde, nicht mehr verfügbar, und sein Nachfolger ist anders aufgebaut. Funktionen müssen erweitert, zusammengeführt oder auf geänderte externe Systeme angepasst werden. Die Pflege oder Weiterentwicklung eines solchen Systems wird für den Produktverantwortlichen zu einer schwer kalkulierbaren Kostenquelle - vor allem dann, wenn er nicht mehr auf den Kern des ursprünglichen Entwicklerteams zurückgreifen kann.

Hilfestellung in solchen Situationen bieten Spezialisten wie Berner & Mattner, die Altsysteme analysieren, bestehende Software-Architektur nachdokumentieren und mögliche Migrations- und Änderungspfade identifizieren. Als externe Architekturberater können sie unbelastet von der Softwarehistorie untersuchen, wo in



Vorteile:

Frühzeitig in die explizite Gestaltung der Software-Architektur zu investieren spart langfristig Kosten für Entwicklung und Wartung.

Ihre Vorteile sind unter anderem:

- Klare Strukturen, professionelle Dokumentation
- Lifecycle-Prognosen
- Integration von State-of-the-Art-Technologien
- Bewusste Designentscheidung auf Basis objektiver Kriterien
- Anpassungsfähige, zukunftssichere Produkte

© iStockphoto.com / Andreas Guskos

Mehr zum Thema erfahren Sie unter: www.berner-mattner.com/sw-architektur

der Struktur der Software Problemfelder liegen, und Lösungen vorschlagen. Der Aufwand für Pflege und Weiterentwicklung lässt sich auf Basis einer solchen Analyse besser beurteilen.

Regeln aufstellen und brechen

Besonders spannend wird das Thema bei zertifizierten Embedded Systemen. Eine häufige Herausforderung ist, dass die Software oder wesentliche Teile zertifiziert sind, um einem SIL (Safety Integrity Level) zu genügen. Selbstverständlich möchte niemand erneut eine IEC 61508-Zertifizierung durchlaufen, wenn sich dies vermeiden lässt. Glücklicherweise sind oft nur einzelne relevante Teile des Systems als sicherheitskritisch eingestuft. Die Kunst des Architekturberaters ist es, die Stellen in der Software zu finden, an denen Schnitte angesetzt werden können zwischen sicherheitskritischen Teilen und den Teilen und Schnittstellen, die problemlos geändert werden können. Hierzu nutzt er gängige Design-Patterns, wie etwa Schichten oder "pipes and filters". Ironischerweise muss häufig das auf diese Weise entstandene Regelwerk – die abgeleitete Architektur für das eigentliche Entwicklungsprojekt – gleich wieder gebrochen werden, um eben zertifizierte Bestandteile unverändert zu erhalten. Der große Unterschied zu den "Regelverletzungen" aus der Entstehungszeit der ursprünglichen Software ist aber der, dass sie nun ausführlich dokumentiert werden.

Software-Architektur als Prozess

Es wäre ein Missverständnis, Software-Architektur als ein unveränderliches Dokument zu verstehen, das man nur selten wieder in die Hand nimmt. Es sollte bei jeder Arbeit an der Software gelesen und gepflegt werden.

Ohne dieses lebendige Dokument werden Architekturverstöße oft gar nicht bemerkt. Vereinbarungen zu Architektur, Coding Conventions, Schnittstellen etc. werden dadurch für den Entwickler verbindlich. Der Auftraggeber erhält so ein Embedded System, dessen Pflege er ohne große Erklärung oder Einarbeitung in die Hände eines neuen Teams geben kann. Die nun freigeordneten Spezialisten, die es erstellt haben, können getrost an anderer Stelle eingesetzt werden. Dabei gilt: Software-Architektur endet nie! Denn auch von den Entwicklern einzusetzende Design-Methoden wie zum Beispiel Object-Oriented Analysis and Design (OOAD) haben klare Strukturen und die Trennung von Zuständigkeiten zum Ziel: Dies ist eindeutig Architektur!

Eine klare Software-Architektur hilft, die Zusammenhänge innerhalb eines Systems besser zu verstehen. Dies verhindert Fehler und senkt den Aufwand bei der "Altbausanierung" von Embedded Systemen. Wurde die Software-Architektur in der Embedded-Entwicklung vernachlässigt, lohnt sich angesichts langer Einsatz- und Lebenszeiten vieler Embedded Systeme auch die nachträgliche Analyse und

Unter Software-Architektur versteht man die strukturierte Anordnung der Komponenten eines Systems unter Angabe der Kommunikationsbeziehungen zwischen den Komponenten sowie deren Abbildung auf Hardware oder Software-Ressourcen.

Dokumentation. Andernfalls drohen immer wieder unkalkulierbare Verzögerungen, wenn Wartungs- oder Weiterentwicklungsarbeiten auf undokumentierte Eigenheiten des Systems stoßen. Bei stark eingefahrenen Teams bzw. zum Wissensaufbau leistet hier ein externer Softwarearchitekt unschätzbare Hilfe.

Die meisten Fehler durch menschliches Versagen entstehen dort, wo wir nicht verstehen, woran wir "schrauben". Die explizite Architektur trägt maßgeblich dazu bei, die Wirkketten innerhalb komplexer Systeme transparent zu machen. Mit diesem Wissen werden erforderliche Änderungen wesentlich schneller bewertet und zuverlässig realisiert. Frühzeitig in die explizite Gestaltung der Software-Architektur zu investieren spart zudem langfristig Kosten für Entwicklung und Wartung.



Elektromobilität

Embedded Software in der Ladeinfrastruktur

Berner & Mattner verfügt über umfangreiches Know-how im Bereich der Elektromobilität, u. a. bei der Schnittstelle zum Auto, beim Hybrid-Powertrain sowie bei Fahrerassistenz- & Telematiksystemen.

Aufgabenstellung

Ausgehend von der Anforderungsanalyse werden die Schnittstellen zwischen Ladeinfrastruktur und externen Systemen definiert. Der modellbasierte Ansatz von Berner & Mattner bietet hier den Vorteil, mit Hilfe von bereits lauffähigen Modellen die Abläufe sehr früh während der Entwicklung gemeinsam mit den Kunden zu verifizieren. Diese Modelle erleichtern auch den wichtigen übergreifenden Austausch mit der Fahrzeugseite.

Das Thema Elektromobilität ist in aller Munde – der flächendeckende Einsatz von Elektrofahrzeugen steht allerdings erst am Anfang. Fest steht: Die Ladeinfrastruktur bildet einen ausschlaggebenden Baustein für den kommerziellen Erfolg des Elektroantriebs. Entscheidend für die breite Akzeptanz sind dabei eine intuitive Bedienbarkeit der Ladesäulen sowie die Einbettung der Ladeinfrastruktur in ein ganzheitliches Konzept, welches dem Anwender einen echten Mehrwert, z. B. beim Parken, bietet.

Intuitive Bedienung der Ladesysteme

Bei der Entwicklung von Software für die Ladeinfrastruktur sind zwei Faktoren ausschlaggebend:

- Unabdingbar ist ein intuitiver Workflow-orientierter Bedienablauf mit einer modernen Bedienoberfläche.
- Das Design der Oberfläche soll austauschbar sein ohne Eingriff in die Funktionalität, um Anforderungen des Endbetreibers der Ladesäule erfüllen zu können.

Zentralsoftware: Frühe Lasttests sind entscheidend

Die Herausforderungen bei der Entwicklung der Zentralsoftware sind anders gelagert: Die Basis bildet eine erweiterbare und auf moderne internetbasierte Technologien ausgerichtete Software-Architektur, die ein dynamisches Anbinden von Adaptern zu externen Systemen ohne Beeinträchtigung des laufenden Gesamtsystems unterstützt.

Der Aufbau von automatisierten Testumgebungen durch Simulation der Hardware (SiL - Software in the Loop) ermöglicht die parallele Entwicklung

von Soft- und Hardware sowie die frühe Durchführung von Lasttests zum Nachweis, dass die Zentrale die zukünftig erwartete Anzahl von Ladebewegungen auch bewältigen wird.

Moderne Berner & Mattner-Technologie

Berner & Mattner setzt auf moderne und bewährte Technologien zur Entwicklung von Bedienoberflächen wie zum Beispiel WPF und XAML und erreicht damit klar definierte Schnittstellen zwischen Design und Funktionalität. Von externen Designfirmen erstellte Oberflächen werden direkt – praktisch in Echtzeit – in die Produktivsoftware übernommen. Für die Entwicklung und Tests werden zuverlässige Continuous Integration Plattformen mit automatisierten Modul-, Integrations- und Systemtests eingesetzt (z. B. VMWare, Jenkins, NUnit). Nicht nur Testfälle innerhalb der Businesslogik werden abgedeckt, sondern auch automatisierte GUI-Tests. Grundlage dafür ist die Berücksichtigung der Testbarkeit schon bei der Erstellung der Software-Architektur (Design for Testability) und das strikte Einhalten von Designpatterns wie MVVM (Model-View-View-Model).



Motorsteuerungen / -applikationen

Requirements Engineering

Hybridsysteme, neue Emissions- und Verbrauchsgrenzen und hochverfügbare Systemintegrationen – die führenden Großmotorenhersteller stehen vor zahlreichen Herausforderungen. Zuverlässig vernetzte Elektronik und modulare Software bilden wichtige Lösungsbestandteile und entscheiden über die Positionierung der Systeme im Wettbewerb. Berner & Mattner unterstützt mehrere Weltmarktführer bei ihren System- & Software Engineering-Prozessen, beginnend mit der Requirements-Analyse.

Systems Engineering heißt Anforderungen festlegen, Systemstrukturen beschreiben, interne sowie externe Schnittstellen fixieren – und dies in der Sprache unserer Kunden. Damit wir uns mit Sicherheit richtig verstehen und das richtige System sowie entsprechende Testfälle entwickeln.

Motoren in zivile sowie militärische Anlagen und Systeme zu integrieren erfordert die Einhaltung umfangreicher, branchen- und länderspezifischer Standards. Berner & Mattner analysiert diese Standards – oft mehrere tausend Seiten starke Werke – und verdichtet die Inhalte auf relevante Kernaussagen mit Priorität:

- Die elementaren Forderungen werden konkret und gut lesbar spezifiziert.
- Lange Passagen werden auf wesentliche Inhalte reduziert.
- In den Vorschriften oft "nur" unauffällige Einzeiler, die es in der Umsetzung aber "so richtig in sich haben", werden systematisch aufgedeckt.

In der anschließenden Gap-Analyse identifizieren wir die wesentlichen Abweichungen in den Elektronik- und Softwaresystemen unserer Kunden hinsichtlich dieser Normforderungen.

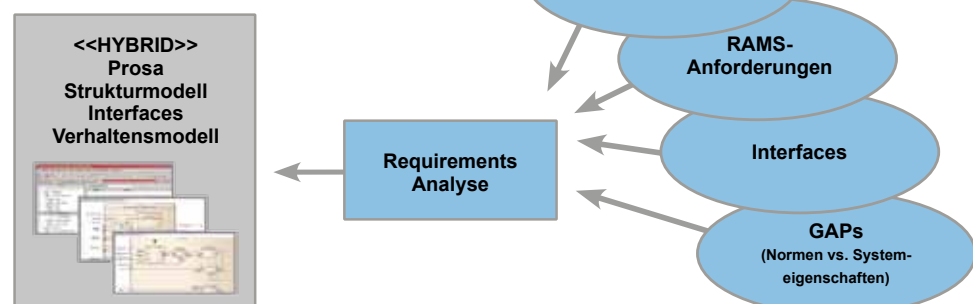
Anforderungs- und FUSI-Analyse

Berner & Mattner setzt für das Requirements Engineering industriebewährte Verfahren ein: z. B. Use Cases, Szenarien, strukturierte Analyse, SysML, UML, um wesentliche Funktionen wie Regler, Turbo-Charger-Modi, Diagnose, Startup-Verhalten, Notbetriebe, etc. zu beschreiben. Eine besondere Bedeutung hat die Analyse der externen und internen Schnittstellen sowie die Durchführung der funktionalen Sicherheitsanalyse (FUSI). Berner & Mattner erreicht mit dem RAMS-Modell – Reliability, Availability, Maintainability und Safety – eine integrale Sicht und leitet daraus ausgewogene System- und Softwareanforderungen ab.

HYBRID-Spezifikationen

Hybrid ist nicht nur "in", sondern außerordentlich wirkungsvoll - sowohl bei Antrieben als auch bei der Systemspezifikation. Unsere Experten setzen für die hochwertigen Spezifikationen eine Kombination aus textbasierter und modellbasierter Beschreibung ein.

Unsere Kunden profitieren von der erzielten Eindeutigkeit bei gleichzeitig guter Verständlichkeit und einer systematischen Traceability zu Test und Design.



Mehr zum Thema erfahren Sie unter: www.berner-mattner.com/motorsteuerungen



Embedded Systems in Space

"Auf zu neuen Wegen" durch Spin-Ins aus anderen Branchen

Erfahrungen von Berner & Mattner mit Embedded Software aus anderen Branchen liefern wichtige Beiträge (Spin-Ins) für die Entwicklung künftiger On-Board Software von Satelliten.

Nicht nur bei der Erforschung der Erde, unseres Sonnensystems und der kosmologischen Zusammenhänge im Universum begibt sich die Europäische Raumfahrt Agentur ESA auf neue Wege. Auch bei der Weiterentwicklung der verwendeten Technologien spielen Erkenntnisse aus anderen Industriezweigen – sogenannte "Spin-Ins" – eine immer wichtigere Rolle.

Elektronik & Embedded Software

Der Anteil an Elektronik und Embedded Software in Satellitensystemen steigt beständig mit deren Komplexität und ist in zunehmendem Maße entscheidend für den Erfolg einer Mission. Jedoch ist die historisch gewachsene Praxis, Elektronik und die dazu passende Software für geplante Missionen jeweils individuell zu entwickeln, mittlerweile nicht mehr zeitgemäß. Ein großer Teil der SW-Funktionalitäten vergleichbarer (Sub-)Systeme nähert sich einander an. Dies eröffnet einen Spielraum für mehr Effizienz – sowohl bei der Entwicklung dieser Software als auch bei deren Qualitätssicherung.

Spin-In für die Raumfahrtindustrie

Vor dem Hintergrund des ständig wachsenden Anteils von Elektronik und Embedded Software kann also

die Raumfahrtindustrie zunehmend von Erfahrungswerten (Best Practices) aus vergleichbaren Industriezweigen profitieren.

Bei der Entwicklung hochkomplexer Satellitensysteme kann sich die Europäische Raumfahrt Agentur ESA daher beispielsweise von Erfahrungen, wie sie in der Automobilindustrie oder im Bereich anderer Transportsysteme gemacht wurden, einen großen Vorteil verschaffen. So werden Erkenntnisse, die bei der Entwicklung des AUTOSAR-Standards oder bei der herstellerübergreifenden Nutzung von Kommunikationsstandards gemacht wurden, in aktuellen Untersuchungen den Erkenntnissen aus vergangenen und laufenden Satellitenentwicklungen gegenübergestellt. Erste Resultate aus solchen Untersuchungen hatten zur Folge, dass die ESA mittlerweile eine Initiative ge-

startet hat, welche die Vereinheitlichung der Software-Architektur für On-Board-Software zum Ziel hat.

Mit Erfahrung auf neuen Wegen

Berner & Mattner verfügt über langjährige Erfahrung in der Entwicklung von Embedded Systemen aus mehreren Industriezweigen und im sicherheitskritischen Umfeld. Dadurch kann das Unternehmen auch entscheidende Impulse und Spin-Ins für die anstehenden Veränderungen im technologischen Fundament der weiteren Entwicklungen im Bereich der Raumfahrt liefern. Auch "Rocket Science" ist also offen für Impulse aus unseren Erfahrungsbereichen! Diese bringen wir in aktuellen Gesprächen ein und tragen so dazu bei, eine solide Grundlage für zukunftsweisende Entwicklungen zu schaffen.

Verteilte Embedded- Bedieninterfaces

Konsistente HMIs für Defence Electronics

Software-Großprojekte – insbesondere im Defence-Bereich – werden standort- und oft länderübergreifend realisiert. Die Abstimmung und Entwicklung einer konsistenten Bedienlogik für das Front-End von Embedded Steuerungen gestaltet sich anspruchsvoll. Berner & Mattner hält für solche Aufgabenstellungen eine bewährte Architektur- und Technologieplattform bereit und unterstützt damit Kunden im internationalen Workshare.

Als Basis für eine verteilte HMI-Entwicklung hat sich bei Berner & Mattner die "Eclipse Rich Client Platform" hervorragend bewährt. Zahlreiche Embedded Front-Ends wurden damit bereits erfolgreich entwickelt.

Bei der Entwicklung komplexer Defence-Systeme ist die architektonische Aufteilung der Komponenten in einzelne Schichten – mit HMI als Schnittstelle zum Nutzer – ein bewährter Weg, um die Komplexität beherrschbar zu machen.

Für komplexe HMIs bietet sich diese Schichtenbetrachtung auch intern an. Die verteilte Entwicklung der Bedienoberfläche in mehreren Teams macht eine Abstraktion der eigentlichen Bedienlogik von Kommunikationsmechanismen sogar zwingend notwendig.

Die Basis bildet eine gemeinsame Kommunikationsschicht. Diese ist spezifisch zugeschnitten auf die systemweiten Kommunikationsmechanismen aber generisch im Bezug auf die transportierten Daten.

Diese gemeinsame Kommunikationsschicht bringt entscheidende Vorteile in der Entwicklung:

- Die einzelnen Komponenten (trotzdem häufig noch sehr komplex) lassen sich hinsichtlich ihrer spezifischen Bedienlogik und Datenströme bearbeiten.

- Die einzelnen Logik-Komponenten werden fachlich separiert und damit die Abhängigkeiten der Komponenten untereinander minimiert.

Ein weiteres Schlüsselkonzept von Berner & Mattner stellen die wiederverwendbaren, graphischen Basisbausteine (Widgets) dar. Sie bilden die technologische Klammer, und das Entwicklerteam erreicht damit ein weitgehend einheitliches Look & Feel. Eingabefelder mit konsequenter Validierung sind hierfür ein gutes Beispiel, aber auch komplexe Tabellen werden somit vereinheitlicht. Dieser integrierte Ansatz ermöglicht es, die fachli-

chen Anteile der Benutzeroberfläche von verteilten Entwicklungsteams implementieren zu lassen, während die wichtigen Usability- und Wartungsanforderungen gewährleistet bleiben.

Schlüsselkompetenzen

- Prototypische und iterativ-inkrementelle HMI-Entwicklungsprozesse
- Systematische automatisierte Tests und Continuous Integration
- Militärische Normen (MIL-2525, MIL-1477, MIL-1472)
- Fundierte und langjährige Erfahrung im Einsatz der Eclipse RCP





DB Netz AG

Hochwertige Schnittstellenspezifikationen durch SysML-Modellierung

Nutzen der Modellierung mit SysML

- präzisere Anforderungen
- erhöhte Vollständigkeit
- höhere Fehlerfreiheit
- bessere Verständlichkeit

Berner & Mattner hat SysML in vielen Projekten mit sehr guten Ergebnissen eingesetzt. Eines dieser Projekte ist NeuPro der DB Netz AG.

Die Entwicklung, Spezifikation und Standardisierung von Schnittstellen zwischen Teilsystemen ist in vielen Branchen, in denen eingebettete Systeme eingesetzt werden, gefordert. Berner & Mattner hat im Rahmen des Projekts NeuPro der DB Netz AG eine Vorgehensweise entwickelt, um qualitativ hochwertige Schnittstellenspezifikationen modellbasiert mit SysML zu entwickeln. Die Methodik konnte im Rahmen der dreijährigen Entwicklungszeit reifen und hat sich als bestens geeignet für komplexe Vorhaben mit einer großen Mitarbeiterzahl erwiesen. Sie beruht auf modernen Konzepten des Systems Engineerings wie System of Systems (SoS) und ist Bestandteil eines Gesamtprozesses zur Entwicklung komplexer Systeme mit SysML.

Beim Einsatz von eingebetteten Systemen wird derzeit in vielen Branchen der Wunsch laut, zwischen Teilsystemen standardisierte Schnittstellen zu definieren und einzuführen. Ziele: Schaffung eines Marktes, Förderung des Wettbewerbs, Senkung der Kosten und Erhöhung der Qualität. Als Beispiel derartiger Initiativen sei der AUTOSAR-Standard im Automotive-Bereich genannt. Im Bereich der Bahntechnik hat Berner & Mattner im Rahmen des Projekts NeuPro für die DB Netz AG ein modellbasiertes Verfahren entwickelt, um Schnittstellenspezifikationen zwischen dem Stellwerk und Feldelementen (wie z. B. einem Lichtsignal) durch die Modellierung mit SysML zu entwickeln.

fifikationen zwischen dem Stellwerk und Feldelementen (wie z. B. einem Lichtsignal) durch die Modellierung mit SysML zu entwickeln.

Vorgehensweise

Die Vorgehensweise ist in zwei Phasen aufgeteilt:

- Modellierung der fachlichen Ebene
- Modellierung der technischen Ebene

Die fachliche Ebene ist eine funktionale, logische und abstrakte Sicht auf die

Anforderungen, unabhängig von speziellen Lösungskonzepten. Lösungsbezogene Anforderungen wie physikalische, elektrotechnische oder softwaretechnische Beschreibungen werden in der technischen Ebene behandelt. Das Systems Engineering bezeichnet die Arbeiten an der fachlichen Ebene als Analysephase.

Die technische Ebene setzt die Anforderungen der fachlichen Ebene mit einem technischen Lösungskonzept um. Diese Phase bezeichnet man im Systems Engineering als Designphase.

Aktivitäten auf fachlicher Ebene

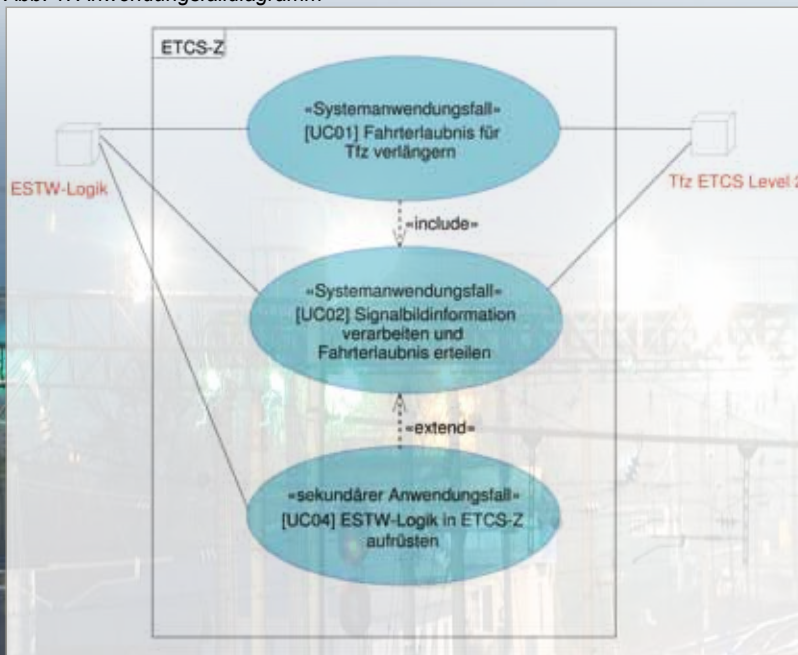
Zu Beginn gilt es festzustellen, welche Teilsysteme des Gesamtsystems Leit- und Sicherungstechnik an der Schnittstellenkommunikation beteiligt sind (im Folgenden als Schnittstellenendpunkte bezeichnet) und welche Teilsysteme bezüglich der zu spezifizierenden Schnittstelle eine Rolle spielen. Dies bezeichnet man als Schnittstellenkontext (Umgebung der Schnittstelle). In der SysML kommt zu diesem Zweck ein Blockdefinitionsdiagramm zum Einsatz. Es dient der Darstellung der statischen Struktur von Elementen (in SysML: Blöcken) und deren Beziehungen (Assoziationen) untereinander.

Basis für die Definition einer Schnittstelle ist im Anschluss die Festlegung, welche Funktionalität auf welchem Teilsystem des Schnittstellenkontextes ausgeführt wird und welche dieser Funktionalitäten eine Kommuni-

kation über die Schnittstelle erfordert. Zur Ermittlung dieser Funktionalitäten wird eine Anwendungsfallanalyse durchgeführt. In der SysML geben Anwendungsfälle die Funktionalität eines Teilsystems in Form von Diensten wieder, die ein Teilsystem für ein anderes Teilsystem oder eine menschliche Person (Akteure) anbietet. Nach Festlegung des Schnittstellenkontextes müssen somit die für die Schnittstelle relevanten Anwendungsfälle identifiziert und in einem Anwendungsfallendiagramm (Abb. 1) modelliert werden. Ein Anwendungsfall besteht aus einem Ablauf von Aktionen, die in wechselnder Abfolge vom Akteur – der den Anwendungsfall nutzt – und vom Teilsystem durchgeführt werden. Das Aktivitätsdiagramm der SysML modelliert diesen Ablauf abwechselnder Akteur-Teilsystem-Aktionen. Es wird im Detail festgelegt, welche Teilfunktionalität auf welchem Teilsystem in welcher Reihenfolge auszuführen ist.

Sind die Abläufe der Anwendungsfälle erarbeitet und die funktionale Verteilung auf die Teilsysteme fixiert, wird in einem nächsten Schritt innerhalb von Sequenzdiagrammen (Abb. 2) die Grundlage für die funktionale Spezifikation gelegt. Aus den erarbeiteten Aktivitätsdiagrammen kann für Regelabläufe (erfolgreicher Standardablauf eines Anwendungsfalles ohne Störungen) die notwendige Kommunikation auf der Schnittstelle durch funktionale Nachrichten (Kommandos und Meldungen) abgeleitet werden. Sequenzdiagramme eignen sich für die Darstellung der Interaktion einiger ausgewählter Kommunikationsszenarien, nicht aber für eine vollständige Beschreibung aller Abläufe. Aus diesem Grund kommt ein weiterer Diagrammtyp ins Spiel: Mit Hilfe von Zustandsdiagrammen für die Teilsysteme der Schnittstellenendpunkte lassen sich nun auch alle Ausnahmefälle erfassen und modellieren, so dass auf fachlicher

Abb. 1: Anwendungsfallendiagramm





Ebene ein hoher Grad an Vollständigkeit erreicht werden kann. Schließlich bieten Zustandsdiagramme die Möglichkeit der Ausführung und Simulation, um eine weitergehende Optimierung und Tests der Anforderungen zu ermöglichen.

Aktivitäten auf technischer Ebene

Den Abschluss bildet die Beschreibung der konkreten technischen Umsetzung. Dies beinhaltet zunächst die Beschreibung der ISO/OSI-Schichten. Kommen Industrieprotokolle und -standards wie Ethernet und TCP/IP zum Einsatz, genügt ein Verweis auf deren Spezifikation. Zuletzt definiert man die technischen Datentelegramme inklusive der Beschreibung ihrer Bytes und Bits sowie ihre Wertebereiche und Bedeutungen.

Berner & Mattner SysML-Prozess

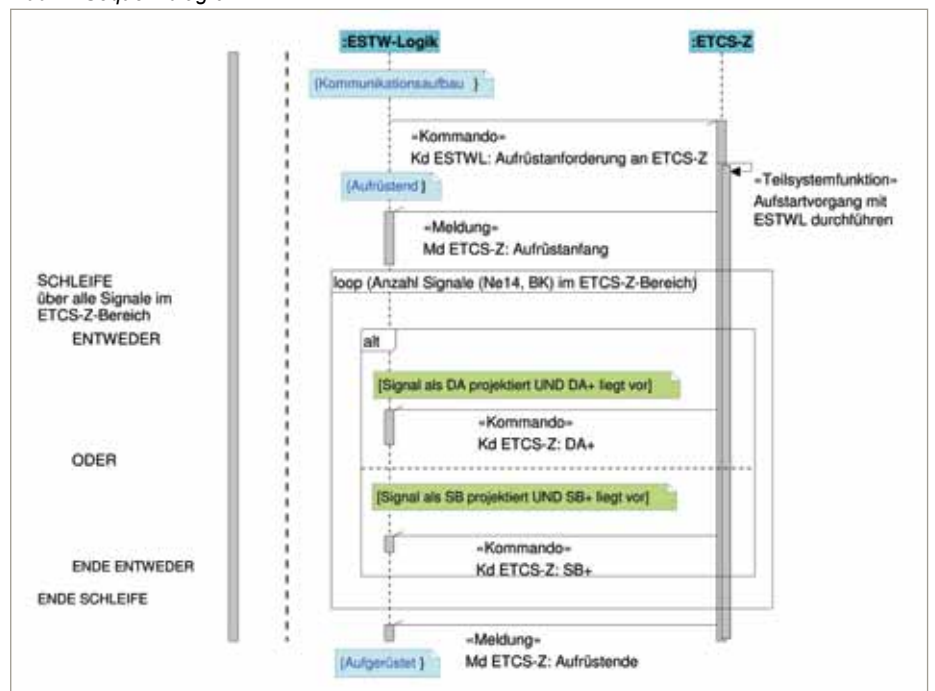
Der Prozess zur Entwicklung von Schnittstellenspezifikationen ist Bestandteil eines umfassenden, von Berner & Mattner entwickelten Prozesses zur Modellierung und Entwick-

lung von bahntechnischen Systemen mit SysML.

Der Gesamtprozess

- basiert auf den Vorgehensmodellen SYSMOD (Systems Modeling Process) und OOSEM (Object-Oriented Systems Engineering Method)
- setzt das Konzept System of Systems (SoS) um
- definiert eine Modellstruktur (umgesetzt durch SysML-Pakete)
- unterstützt die Modellierung von Varianten
- integriert die Phasen des RAMS-Lebenszyklus
- unterstützt drei Ebenen der Darstellung von bahntechnischen Anforderungen (betriebliche / fachliche / technische Ebene)
- erleichtert die Verständlichkeit durch Einschränkung der Anzahl der eingesetzten Modellelemente der SysML, Vorgabe von Modellierungsrichtlinien, Integration und Bevorzugung von bahntechnischen Termini sowie weitgehende Darstellung in deutscher Sprache.

Abb. 2: Sequenzdiagramm



Software-Standards für Satellitenanwendungen

Entwicklungshandbuch für effiziente Umsetzung

Software, welche für die Steuerung im Satelliten oder aber auch zur Steuerung des Satelliten am Boden eingesetzt wird, muss höchste Anforderungen erfüllen. Korrektheit und Zuverlässigkeit sind die Kernpunkte, um eine erfolgreiche Durchführung der Satellitenmission sicherzustellen. Daher wird an die Entwicklung der Software ein sehr strenger Maßstab angelegt und der Entwicklungsprozess sehr genau vorgegeben.

Das Entwicklungshandbuch als zentrales Nachschlagewerk ist ein zuverlässiges Hilfsmittel, um den hohen Anforderungen an die Entwicklung von Satellitensoftware zu entsprechen.

Die Europäische Raumfahrt Agentur (ESA) hat die Anforderungen für die Entwicklung ihrer Projekte in ECSS-Standards definiert. In diesen Standards werden unterschiedliche Aspekte der Entwicklung in verschiedenen Teildokumenten behandelt. Für die Software-Entwicklung sind z. B. die Standards zum Engineering und für die Qualitätssicherung der Software zu beachten, ebenso aber auch die allgemeinen Standards zum Projektmanagement oder zum Systems Engineering.

Der Entwickler von Software für Satellitenanwendungen muss also eine Vielzahl unterschiedlicher Dokumente kennen, um alle Anforderungen bei der Entwicklung zu berücksichtigen. Darüber hinaus ist für das konkrete Projekt ein Tailoring durchzuführen, bei dem in Abhängigkeit von der Kritikalität der Anwendung die konkret durchzufüh-

renden Maßnahmen im Detail festgelegt werden.

Um den Entwicklern von Satellitensoftware ein zuverlässiges Hilfsmittel für die Anwendung der geforderten Prozesse an die Hand zu geben, empfiehlt sich die Entwicklung eines Handbuchs, in dem die konkreten Forderungen und Maßnahmen definiert sind. Dies bietet verschiedene Vorteile:

- Zentrales Nachschlagewerk für alle relevanten Anforderungen an den Entwicklungsprozess
- Einheitliche Vorgaben für das Prozess-Tailoring
- Einheitliche Definition von Methoden zur Umsetzung der Entwicklungsprozesse
- Klärung missverständlicher oder interpretationsbedürftiger Formulierungen aus den Standards

Diese Vorgehensweise wurde z. B. im Galileo-Projekt umgesetzt. Ein Entwicklungshandbuch hilft zudem dabei, bei einer verteilten Entwicklung an verschiedenen Standorten und durch unterschiedliche Unternehmen effizient eine einheitliche Vorgehensweise umzusetzen.

Berner & Mattner unterstützt seine Kunden bei der Einführung und Umsetzung anspruchsvoller Entwicklungsstandards mit der Entwicklung von Prozesshandbüchern, der Definition und Einführung von Methoden sowie der Umsetzung in entsprechende Tools. Dabei finden die Standards für die Entwicklung sicherheitskritischer Systeme aus verschiedenen Branchen Anwendung. Darüber hinaus entlasten wir Kunden aus Industrie & Wissenschaft durch die effiziente und zuverlässige Übernahme von Produktsicherungs- und Projektmanagementaufgaben.

Industrial Embedded Systems

Berner & Mattner auf der embedded world 2012



Anfang Februar präsentieren wir auf der embedded world 2012 unser Leistungsspektrum aus dem Bereich "INDUSTRIAL EMBEDDED SYSTEMS", in dem wir bewährte Best Practices besonders nutzbringend vereinen und unsere branchenübergreifenden Synergien noch gezielter nutzen können.

Zusammen mit unseren Kollegen und Kolleginnen der französischen Assystem Group stellen wir unsere gemeinsame Engineering- und Consulting-Kompetenz für den Lifecycle von elektronischen Steuergeräten und

Steuersystemen vor – von der Spezifikation über Entwicklung und Testautomatisierung bis zur Serienbetreuung.

Wir freuen uns, wenn Sie sich den Termin 28.02. bis 01.03.2012 in Ihrem

Kalender vormerken und uns am Stand 326 in Halle 5 besuchen. Anhand mehrerer Exponate stellen wir Ihnen innovative kundenspezifische Lösungen und interessante Geschäftsmodelle vor.

Kurznachrichten

>> Nächste Veranstaltungen

Berner & Mattner präsentiert seine Leistungen auf dem Kongress Embedded Software Engineering (06. - 08.12.2011) in Sindelfingen. Besuchen Sie uns!
Info: www.berner-mattner.com/ese2011

Ebenso ist Berner & Mattner auch im nächsten Jahr auf der embedded world in Nürnberg vertreten (28.02. - 01.03.2012). Wir freuen uns über Ihren Besuch an unserem Stand 326 in Halle 5.
Info: www.embedded-world.de

>> Vortrag ESE-Kongress

Dr. Michael Sturm von Berner & Mattner referiert am 07.12.2011 in Sindelfingen zum Thema "Architecture Lifecycle Optimierung mit Embedded Design Pattern".
Info: www.ese-kongress.de

>> Vortrag ERTS 2012

Auf der Embedded Realtime Software and Systems in Toulouse (01. - 03.02.2012) referiert Berner & Mattner zum Thema "Requirements and Test Case Tracing".
Info: www.erts2012.org

>> Fachbeitrag "Funktionsorientierte Spezifikation als Vertrag zwischen Auftraggeber und Hersteller"

In diesem Artikel wird eine Spezifikationstechnik zur Beschreibung von Systemchnittstellen vorgestellt, die heute bei der Berner & Mattner Systemtechnik GmbH erfolgreich eingesetzt wird.
Info: www.berner-mattner.com/fachartikel

>> Fachbeitrag "Sicherheitsnormen als Mittel zur Qualitätssteigerung"

Berner & Mattner beschreibt eine Vorgehensweise zur modellbasierten Entwick-

lung sicherheitskritischer Systeme, die stark an das V-Modell aus der CENELEC-Normenfamilie angelehnt ist.

Info: www.berner-mattner.com/fachartikel

>> Fachbeitrag "Integration von Funktionaler Sicherheit im Entwicklungsprozess"

Berner & Mattner stellt eine neue Methodik vor, die Sicherheitsaktivitäten werkzeuggestützt mit der klassischen Systementwicklung verzahnt und Sicherheitsanforderungen frühzeitig ins Systemdesign einbezieht.

www.berner-mattner.com/fachartikel

>> Neuigkeiten aus der Produkthecke

Ab sofort ist die Version 3.3 der Simulations- und Testplattform MESSINA verfügbar. Neben einem GUI-Redesign wurden die Geschwindigkeit des Systems und die zeitliche Messgenauigkeit erhöht.



**Zum Ausbau unseres Teams suchen wir für
den Bereich Industry in München (m/w):**

Softwarearchitekt

Projektleiter Softwareentwicklung

Softwareentwickler Embedded Software

Softwareingenieur .NET Industrieautomation

**Softwareentwickler Java - 3D Visualisierung
mit OpenGL**

**Entwicklungsingenieur Sicherheitskritische
Systeme**

Insbesondere Hochschulabsolventen und Quereinsteigern (m/w) bieten wir ein intensives und individuelles Einarbeitungsprogramm. Berner & Mattner bietet attraktive Konditionen, eine individuelle Förderung, Unterstützung aus unseren Teams, Freiraum zur Gestaltung Ihrer persönlichen Karriere und die Möglichkeit, Verantwortung zu übernehmen.

Nichts für Sie dabei?

Wenn Sie Erfahrung in der Entwicklung technischer Software haben und Kenntnisse im Bereich Industrieautomation oder aber Automobil, Luft- und Raumfahrt, Verteidigung oder Schienenverkehr mitbringen, freuen wir uns auch auf Ihre Initiativbewerbung. Wir prüfen Ihre Unterlagen und nehmen Kontakt mit Ihnen auf, sobald wir eine geeignete Position zu besetzen haben.

Detaillierte Informationen zum Unternehmen und den einzelnen Positionen finden Sie auf <http://www.berner-mattner.com>. Wir freuen uns auf Ihre aussagekräftigen Bewerbungsunterlagen. Bitte nutzen Sie gleich unser Online-Bewerbungsformular unter <http://www.berner-mattner.com/bewerben>.

Frau Anja Ranft
Human Resources Manager
Erwin-von-Kreibitz-Str. 3
80807 München
Tel. (089) 60 80 90-201
Fax (089) 609 81 82
Anja.Ranft@berner-mattner.com

www.berner-mattner.com/jobs



berner & mattner
optimizing your development

Mit Software-Architektur berühmt werden?

Machen Sie es richtig!



Mai 2011

- >> Mitglied von Assystem - Berner & Mattner wird international
- >> Verteiltes Engineering - Schnell, sicher und effizient entwickeln
- >> Softwareentwicklung - Branchenübergreifende Zuverlässigkeit



Oktober 2010

- >> Systems Engineering für unterschiedliche Anwendungen
- >> Partnerschaften mit Industrie und Wissenschaft
- >> Werkzeuge für die Softwareentwicklung



April 2010

- >> Modellbasierte Systemspezifikation in der Luftfahrt
- >> Virtuelle Build- und Testumgebungen in der Produktentwicklung
- >> Innovative Lösungen mit Human Machine Interfaces

Die vorangegangenen Ausgaben des *Insight* Industry finden Sie zum Download unter:

www.berner-mattner.com/de/download-center/newsletter



www.berner-mattner.com

München - Berlin - Ingolstadt - Köln - Stuttgart - Wien - Wolfsburg