

Trend Guide

Embedded Future



MICROCONSULT

No.1 in Developer Training

Dieser Trend Guide wurde gesponsert von:

Microsoft[®]

**Wir schaffen die Plattformen
für Ihre embedded Zukunft.**

**Der Microsoft Beitrag für die anspruchsvollen
Entwicklungen der nächsten Generation:**

Windows CE.Net und Windows XP Embedded

Alle Infos unter:

<http://www.microsoft.com/germany/ms/embedded>



Embedded Future

■ Vorwort	04
Ihre Ansprechpartner bei MicroConsult	
■ Trend Spots	06
Wie Internet-Services den Embedded-Markt aufrollen	
■ Die Zukunft: das universelle Netz	08
Immaterielle Werte hoch im Kurs	
■ Die Chance: Mehrwerte schaffen	12
Neue Dienste, faire Abrechnung	
■ Die Herausforderung:	17
Flexible Architekturen für Unabhängigkeit	
■ Microsoft.Net: Einsatz in der Industrie	26
Visualisierung, verteilte Systeme und Connectivity	
■ Kurse zum Thema	36
Für alle, die es genau wissen wollen	

Vorwort



Liebe Leser,

der Erfolg des Internets beruht auf der Weiterentwicklung von Strategien, die sich durch die gesamte Evolution des Lebens bewährt haben. Dazu gehört einerseits die Bildung vernetzter Systeme, mit denen wir Ressourcen wie Energie oder Information optimal nutzen.

Andererseits dienen diese Netze dem Akkumulieren und effektiveren Verwerten von Wissen. Immaterielle Werte gewinnen in unserer Informationsgesellschaft die Oberhand über materielle Werte.

Das Internet eröffnet hier durch seine relativ einfache und kostengünstige Verfügbarkeit für immens viele Nutzer eine völlig neue Dimension. Es entwickelt sich rasant zu etwas, was wir bei MicroConsult das „Universelle Netz“ nennen – und es wird die Zukunft des Embedded-Marktes maßgeblich bestimmen. Produkte, Leistungen, Methoden, Prozesse und Arbeitsteilung werden sich dadurch stark verändern. Die Weichen dazu müssen alle Beteiligten schon heute stellen.

Das MicroConsult-Team hat diese Problematik intensiv mit Branchenkennern diskutiert und die Kernideen in diesem Trend Guide für Sie zusammengefasst. Erfahren Sie, mit welchen Embedded-

Das MicroConsult-Team hat diese Problematik intensiv mit Branchenkennern diskutiert und die Kernideen in diesem Trend Guide für Sie zusammengefasst. Erfahren Sie, mit welchen Embedded-

Diensten Sie Ihre Kunden langfristig binden und wie Ihre Lösungsarchitekturen offen für neue Anforderungen und Technologien bleiben. In diesem Zusammenhang werden wir die .NET-Strategie und ihre Perspektiven für industrielle Anwendungen beleuchten.

Der Trend Guide informiert Kunden und Interessenten von MicroConsult regelmäßig über aktuelle Themen, Trends und Technologien. Wenn Sie den letzten Trend Guide „Software-Engineering und Internet“ sowie die kommenden Ausgaben kostenlos beziehen möchten oder Anregungen zu Themen und Inhalten haben, schicken Sie uns ein Mail an trendguide@microconsult.de oder ein Fax an +49 (0) 89-45 06 17-18. Falls Sie spezielle Fragen zu unserem Schulungsprogramm haben, wenden Sie sich einfach direkt an uns. Wir beraten Sie jederzeit gerne. Einen Überblick über unsere Kurse zu diesem Trend Guide „Embedded Future“ finden Sie ab Seite 36.

Viel Spaß beim Lesen!



Klaus-Peter Rosenthal

Product Manager Software Entwicklung

E-Mail k.rosenthal@microconsult.de

Ihr MicroConsult-Team

Trend Spots

■ Geschwindigkeitsplus

Das Internet der Zukunft wird Verbindungen 100 mal schneller als heute bereitstellen.

Zwei Projekte führen momentan das Wettrennen um das schnelle Internet von morgen an: „Internet2“ ist ein Konsortium von 180 US-Universitäten, die „Geant“-Initiative verbindet 3.000 europäische Universitäts- und Forschungsinstitute.

■ Integrationsleistung

Eine für alles? Universelle Entwicklungsplattformen sollen zukünftig die Tool-Programmierung über alle Standard- und Prozessgrenzen hinweg vereinfachen. Das Eclipse.org Consortium will beispielsweise eine neue Ära von Entwicklungstools für Applikationen einläuten. Die Eclipse Plattform soll es Tool-Entwicklern ermöglichen, völlig integrierte Tools zu programmieren. Die Plattform arbeitet dabei mit jeder Quelle: Java, C, Word, HTML, JSP und anderen mehr.

■ Verbreitung schafft Masse

Je mehr Geräte an das Internet angebunden sind, desto mehr Services wird es für die Anwender geben.

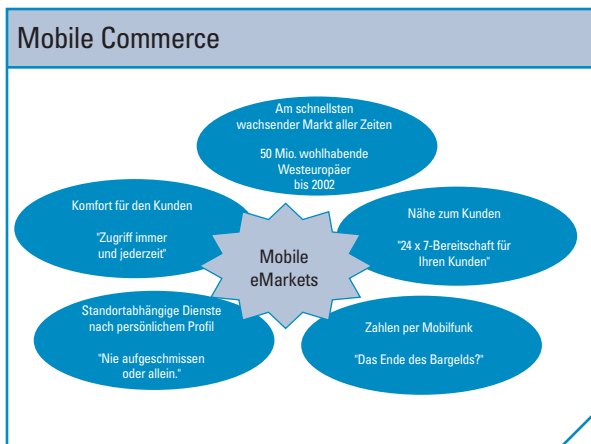
Heute sind an das Internet weltweit 100 Millionen Geräte angeschlossen, im Jahr 2010 sollen es 14 Milliarden sein (Forrester Research 2001). Im nächsten Jahr könnten 100 Millionen Geräte mit Bluetooth-Technik verkauft werden (IDC 2001). Im Jahr 2004 soll es 1 Milliarde Handy-Besitzer geben, 600 Millionen werden das mobile Internet nutzen.

■ Faire Abrechnung

Zukünftig wird der Anwender nur die Leistungen bezahlen, die er in Anspruch genommen hat. Service Provider investieren immense Summen in neue Billing-Lösungen, um vom Flat-Rate-Billing auf nutzungsbasierte Abrechnung umzustellen.

■ Sicherheit ausschlaggebend

Nur ein Anwender, der sich vor Datenmissbrauch sicher fühlt, wird Internet-Services in Anspruch nehmen – und dafür bezahlen. Über die nächsten 3 bis 5 Jahre sollen die Gesamtbudgets für IT-Sicherheit weltweit um 20 bis 40 Prozent jährlich steigen. Grund: die enorm starke Zunahme von E-Business-Initiativen in Unternehmen.



Quelle: TIMElabs

Der größte Wachstumssektor im Geschäft mit internetbasierten Geräten

Die Zukunft: das universelle Netz

Märkte funktionieren, wenn sie menschliche Bedürfnisse erfüllen. In unserer Informationsgesellschaft orientieren sich die Märkte an Wünschen in den Bereichen Sicherheit, Kommunikation sowie Selbstverwirklichung. Was so einfach klingt, hat immense Auswirkungen auf Embedded-Produkte.

Zwei Trends treiben den Homo Sapiens schon seit den Anfängen an und erhalten durch neue technologische Möglichkeiten in der Zukunft eine immer stärkere Ausprägung:

- Trend zu vernetzten Systemen
- Trend zu immateriellen Werten

Diese Tendenzen ziehen sich wie ein roter Faden durch die gesamte Evolution: angefangen bei der Bildung von Zellhaufen über die Entwicklung von Nervensystemen mit zentralen Schaltstellen (Gehirn) bis hin zur Vernetzung intelligenter Wesen durch Kommunikationsmittel. Vernetzte Systeme sind immer eine Kombination aus einem Transport- und einem Kommunikationssystem sowie einem Speichersystem für Energie, Information oder Materie. Sie ermöglichen seit Jahrtausenden eine höhere Spezialisierung und schaffen damit Freiräume für die Verwirklichung größerer Ziele. So basierte schon der Erfolg des Römischen Reiches wesentlich auf dem konsequenten Auf- und Aus-

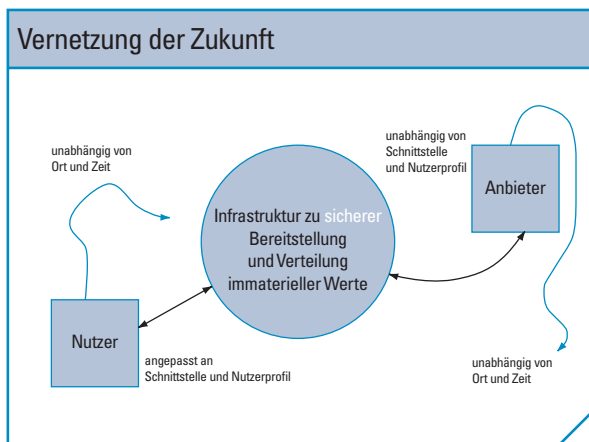
bau von Straßennetzen. Wenn mehrere dieser Netze verschmelzen, also konvergieren, steigt der Nutzen exponentiell. Typisch dafür ist die Kombination von Versorgungs-, Verkehrs-, Transport- und Kommunikationsnetzen. Schon heute ist jeder einzelne in viele konvergente Netze eingebunden. Wer die Netze zählt, die er nutzt, wird über die Zahl staunen und darüber, wie stark sie miteinander verwoben sind. Gleichzeitig erweisen sich viele dieser Netze als äußerst nützlich, obwohl dieser Nutzen keinen materiellen Wert darstellt.

Der zweite Urtrend betrifft also das Erzeugen und Verwenden immaterieller Werte, heute in Form von Software, multimedialen Daten und Informationen oder Dienstleistungen. Bei genauem Hinsehen besteht beispielsweise der eigentliche Wert eines Halbleiterchips zu mehr als 99 Prozent aus Wissen.

Mehrwerte durch Software

Softwarezentrierte Technologien sind eindeutig auf dem Vormarsch. Die Wertsteigerungen vieler Produkte beruhen weniger auf dem Material, aus dem sie hergestellt werden. Im Gegenteil: ihre Qualität und Unterscheidung von ähnlichen Produkten sind Funktionalität, Komfort, Ergonomie, Kommunikationsfähigkeit und Intelligenz – durch Software repräsentiert.

Gefordert ist eine Infrastruktur, die immaterielle Werte wie beispielsweise Musik, Videos, Informationen, Dienstleistungen oder Software (ASP) sicher bereitstellen und verteilen kann. Jeder Anwender soll dabei nach landläufiger Meinung jederzeit und von jedem Ort über alle Übertragungswege wie Funk, Kabel oder Satellit erreichbar sein.



Die Infrastruktur für immaterielle Werte muss dem Anwender den freien und sicheren Zugriff auf Informationen und Services ermöglichen.

Schön und gut, doch welchen Unterschied macht es dann noch, ob römische Diktatoren der Frühzeit oder Technologien der Neuzeit die Menschheit versklaven? Eine berechtigte Frage, die sich konkret beantworten lässt: Jeder Anwender sollte jederzeit und überall frei entscheiden können, wie er welche

Dienste zu welchem Zeitpunkt nutzen will. Technologien, die heute noch dem Business-to-Business vorbehalten sind, wandern morgen in die Privathaushalte, wie beispielsweise mit der Gebäudeüberwachung per Internet. Dabei darf es keine Rolle spielen, ob der Anwender im Auto unterwegs ist, mit dem Handy am Flughafen oder am PC im Büro sitzt. Die Anbieter müssen demnach Dienste und Lösungsarchitekturen bereitstellen, die sich an jede Benutzerschnittstelle und jedes Profil anpassen.

Hochflexible Lösungsarchitekturen

Ein Freund von klassischer Musik möchte beispielsweise über eine neue Aufzeichnung informiert werden und diese eventuell gleich im Auto anhören. Ein anderer Anwender teilt seinem Organizer mit, wann er Privatperson ist und das aktuelle Sportangebot am Reiseort erfahren und wann er sich als Geschäftsmann über die aktuellen Börsenentwicklungen informieren möchte. Diese Anforderungen haben immense Auswirkungen auf die Organisation des Produktumfeldes. Gefragt sind flexible Lösungsarchitekturen (siehe Seite 17), die materielle Produkte mit immateriellen Werten wie Dienstleistungen und Kommunikationsfähigkeit geschickt kombinieren. Die Einbindung von Internet-Technologien steht im Embedded-Bereich allerdings erst am Anfang. Auch das Internet selbst bietet noch lange nicht die Möglichkeiten, die ein universelles, allgegenwärtiges Netzwerk der Zukunft offerieren wird. Eine rasante Entwicklung in diese Richtung zeichnet sich jedoch bereits ab. ■

Die Chance:

Mehrwerte schaffen

Viele Produkte erfahren zukünftig ihre Wertsteigerung durch das universelle Netz vor allem im immateriellen Bereich. Das Gerät ist nicht die eigentliche Leistung, sondern wandelt sich mehr und mehr zum Dienstleistungsterminal für viele geschickt verknüpfte Leistungen.

Das Internet fördert den Trend zu Thin Clients, die schnell den Zugang zu Dienstleistungen ermöglichen, ohne die Leistungsfähigkeit eines klassischen PCs besitzen zu müssen. Mit diesen schlanken Geräten zahlen Anwender den Komfort der Nutzung und mit den Abrechnungseinheiten die Nutzungsintensität von Kommunikationsdienstleistungen.

Internet inklusive

Mehr und mehr Produkte werden heute bereits mit einer Internetschnittstelle ausgeliefert, die eine Kombination mit weiteren Leistungen und gleichzeitig eine nutzungsorientierte Abrechnung ermöglichen. Zu denken ist dabei nicht nur an mobile Devices wie PDAs, Handhelds oder Mobiltelefone und Steuerungs-, Informations- sowie Sicherheitssysteme im Automobil. Auch Privathaushalte werden zukünftig verstärkt in das universelle Netz eingebunden sein, beispielsweise mit webbasierten Heizungs-, Beleuchtungs- oder Überwachungslösungen. Unternehmen im Embedded-Umfeld müssen deshalb detailliert

hinterfragen, welche Dienste sie mit welchem Gerät verbinden, um vermarktbarere Mehrwerte zu schaffen. Die Kernkompetenzen im Produkt oder im Umfeld können dabei ganz unterschiedlich sein, beispielsweise Dienste, die

- den Betrieb sicherstellen,
- die Nutzung fördern,
- die Kundenbindung erhöhen oder
- die Transparenz verbessern.

In diesem Zusammenhang bildet die Abrechnung von Leistungen, das sogenannte Billing, eine zentrale Herausforderung im „Universellen Netz“. Hier wäre statt des simplen Stromzählers ein intelligenter Dienstleistungszähler denkbar, der über das Internet abgerufene Dienstleistungen und Produkte detailliert berechnet.

Die nutzungsgerechte Bezahlung von Musik, Video, Informationen, Kommunikation, aber auch beispielsweise der Verwendungsdauer und -intensität beliebiger Haushalts- und Bürogeräte ermöglicht völlig neue Geschäftsmodelle. In vielen Fällen wird es eine Verschiebung von besitz- zu verbrauchsorientierten Preisgestaltungen geben. Die technologische Basis dafür ließe sich bereits heute kostengünstig realisieren. In vielen Fällen haben sich ähnliche Modelle aus dem Leasing-Geschäft bereits bewährt.

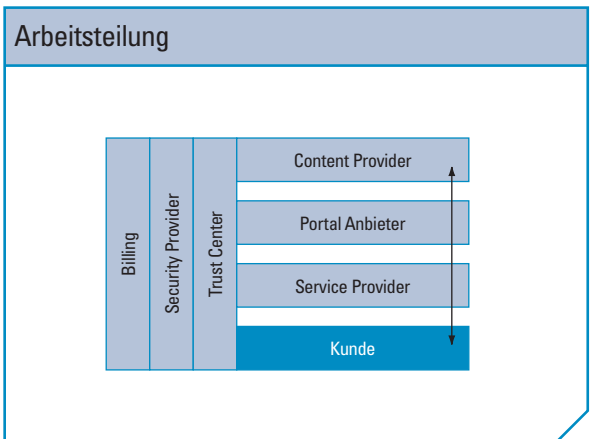
Der heute schon wesentliche Aspekt der umfassenden Sicherheit wird für die erfolgreiche Weiterentwicklung zum „Universellen Netz“ eine Schlüsselrolle spielen. Sie ist der kritische Pfad, der über Akzeptanz und Verbreitung eines „Universellen Netzes“ zur Verteilung und Abrechnung von Leistungen entscheidet. Abgesehen von rein technischen Problemen sind aber noch viele politische, bürokratische und juristische Hürden zu nehmen. Biometrische Verfahren in Kombination mit der Chip-Karte könnten hier Standards setzen. Größtmögliche Sicherheit erfordert allerdings breiten Konsens von Lösungsanbietern, Politik und Rechtsprechung: Nur so lässt sich eine Kombination von

- Schutz der Privatsphäre und vor Datenmissbrauch, zum Beispiel durch Public-Key-Infrastrukturen,
- Schutz vor unseriösen Anbietern und Kunden durch Trust Centers,
- Schutz vor Hackern und Viren durch Firewalls, Virenchecker, u.a. sowohl effektiv als auch wirtschaftlich realisieren.

Viele Unternehmen mit hochspezialisiertem Know-how sind und werden nicht in der Lage sein, all diese Problemfelder selbst ausreichend abzudecken, ohne den Anschluss an das eigentliche Kerngeschäft zu riskieren. Schon mit einem marktgerechten Webauftritt sind viele Unternehmen überfordert. Deshalb besitzen ihre Webseiten eher

das Format eines Tante-Emma-Ladens und taugen kaum dazu, die Vermarktung hochwertiger High-Tech-Produkte zu unterstützen oder das Leistungsspektrum zu erweitern. Ganz zu schweigen von der Realisierung tragfähiger Sicherheitskonzepte.

Wer das Geschäft in Kombination mit den Möglichkeiten eines „Universellen Netzes“ auf tragfähige Beine stellen will, wird sich deshalb zur konsequenten Arbeitsteilung durchringen müssen. Es gilt, die eigene Rolle in diesem übergreifenden Prozess genauestens zu definieren. Unternehmen müssen sich fragen, ob sie als Content Provider Inhalte bereitstellen, als Portal Anbieter diese Inhalte auf



Im „Universellen Netz“ übernehmen Unternehmen Spezialaufgaben wie Abrechnung oder Sicherheit.

einer attraktiven Plattform vermarkten oder als Service Provider die komplexe technische Logistik bieten wollen, welche die Informationen zu den Kunden bringt. Es macht ja auch heute wenig Sinn, eine Telefongesellschaft zu gründen, um telefonieren zu können, oder eine Druckerei einzurichten, um Datenblätter zu drucken.

Spezialisierte Unternehmen werden in den meisten Fällen das Billing, Security Providing und die Zuverlässigkeit der Geschäftsabwicklung (Trust Center) über alle Ebenen hinweg übernehmen. Konvergenz in den Netzen erfordert also einen eindeutigen Konsens zwischen den Anbietern bezüglich gemeinsamer Standards und wirtschaftlicher Arbeitsteilung. ■

Kompetenz bestimmt Produkt

- Wo liegen die entscheidenden immateriellen Werte meines Produktes, Unternehmens?
- Was spiegelt meine Kernkompetenz am besten wider?
 - Der materielle Wert
 - Der immaterielle Wert
- Was soll angeboten werden?
 - Das materielle Produkt
 - Die immateriellen Leistungen
 - Eine Mischung aus beiden

Die Herausforderung: flexible Architekturen

Wer heute den rasanten Wandel der Kundenanforderungen und Technologien konstatiert, erntet meist gelangweiltes Gähnen. Zuviel wurde bereits darüber geschrieben und geredet. Doch sind unsere Lösungsarchitekturen wirklich schon so flexibel, dass sie sich jederzeit dynamisch zukünftigen Änderungen anpassen können? Haben wir wirklich erkannt, wo wir konkret ansetzen müssen?

Kunden werden sich bevorzugt für den Anbieter entscheiden, der sie auf dem Gerät situationsgerecht informiert, das sie gerade verwenden – und dies in geeigneter Darstellung mit sinnvollen Möglichkeiten zur Interaktion. Die daraus resultierende Komplexität und geforderte Flexibilität der Lösungen wird begleitet von zunehmender wettbewerbs- wie technologiebedingter Innovationsgeschwindigkeit.

Dieses Dilemma lässt sich nur durch eine wachsende Entkopplung von Lösungsarchitektur und -technologien in den Griff bekommen. Letztendlich sind wir durch die Schnelllebigkeit der heutigen Produkte gezwungen, Architekturen zu entwickeln, die sich auf Technologien anpassen und erweitern lassen, welche erst in einiger Zeit auf den Markt kommen. Genau im Design solcher Architekturen liegt in der Regel die entscheidende Kernkompetenz.

Die Verifikation der Architektur muss bereits in einem frühen Stadium weitgehend möglich sein,

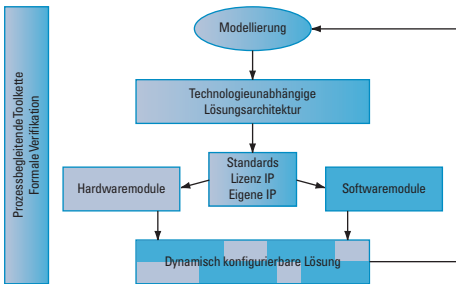
dies ohne den aufwendigen Prozess der Implementierung auf einer Zielhardware. Schließlich haben die wenigsten teuren Probleme mit einer falschen Befehlszeile zu tun, sondern mit der Logik der gesamten Architektur. Und da ist ein völliger Neubeginn der Programmierung nach dem Super-Gau keine Seltenheit.

Diese Vorgehensweise bietet zudem den Vorteil, dass sich aus der Architektur Lösungen mit unterschiedlichen Kombinationen von Hardware- und Softwaremodulen ableiten lassen.

Vorreiter Automobilbranche

Und damit lässt sich die Aufteilung von Funktionalität in Silizium zu Funktionalität in Software unter den Gesichtspunkten Fertigungskosten, verfügbare Lösungsbausteine, Stückzahlen, Energiebedarf, Speicherbedarf, usw. schneller optimieren. Auch die Bildung von Produktfamilien wird erleichtert, die dieselbe Architektur in Produkten mit unterschiedlichem Leistungsumfang, Design oder Bedienkomfort verwenden. Im Automobilbau schon längst Realität, in der Software leider immer noch die Ausnahme. Die Software-Entwickler hätten dabei eigentlich ein leichteres Spiel, aber es fehlt schlicht an vielen Stellen an sauberen Prozessen und der Einsicht, dass Kreativität und die konsequente Umsetzung definierter Prozesse kein Widerspruch sind.

Moderne Architekturen und Prozesse



I-Logix

Modellierung schafft die Unabhängigkeit von der eingesetzten Technologie.

Deshalb wird die Modellierung auch in der Embedded-Entwicklung immer mehr an Bedeutung gewinnen, da sie den Entwurf einer weitgehend technologieunabhängigen Lösungsarchitektur ermöglicht. UML (Unified Modeling Language) und erweiterte Varianten für den Entwurf von Echtzeitsystemen wie z.B. von I-Logix, Artisan und Rational bieten hier bereits heute eine hilfreiche Unterstützung.

Eine weitere Konsequenz der steigenden Komplexität und Flexibilität bei erhöhtem Innovationsdruck ist die Notwendigkeit, verstärkt auf vorgefertigte

Lösungsbausteine zurückzugreifen und Standards zu nutzen oder zu definieren. Lösungsarchitekturen so zu entwerfen, dass sie mit Hilfe standardisierter oder „of-the-shelf“ Komponenten realisierbar sind, wird auch im Embedded-Markt über den entscheidenden Vorsprung bei der Time-to-Market entscheiden. Wer heute das Rad neu erfinden will, ist sicherlich zum Scheitern verurteilt. Spezialisierte Anbieter von Prozessorarchitekturen, Peripherals, Echtzeitbetriebssystemen, Protokoll-Stacks, Treibern oder konfigurierbaren Mensch-Maschine-Schnittstellen haben meist die Innovationsnase vorn und bieten in der Regel die bessere Qualität. Wer Lösungsbausteine offeriert, muss größten Wert auf hohe Integrationsfähigkeit legen, beispielsweise durch nachgewiesene Konformität zu Standards und das Bereitstellen von UML-Modellen. Die Anwender von Lösungsbausteinen müssen lernen, die Qualität dieser Module zu definieren und zu prüfen.

Bausteine und Edelsteine

Eigenentwicklungen sollten da zum Einsatz kommen, wo sich alles um die Kernkompetenzen des Unternehmens dreht, denn sie sind die Edelsteine, mit denen sich ein Produkt von anderen wirklich unterscheidet. Dies wird umso notwendiger, je vielfältiger und flexibler die Nutzungsmöglichkeiten vernetzter Lösungen durch die Unabhängigkeit von Ort und Zeit werden. Gleichzeitig soll sich die Lösung an die Fähigkeiten und Erfordernisse der

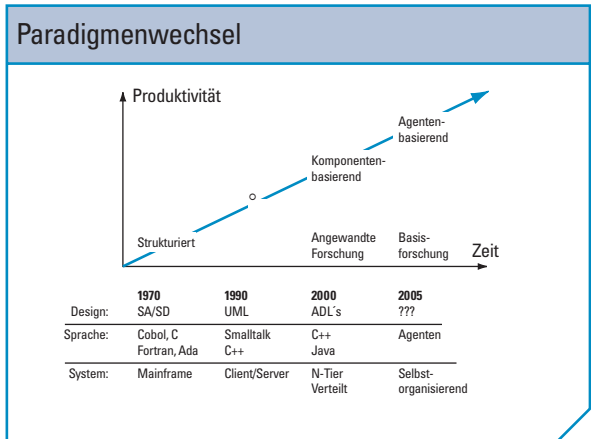
Alleinstellungsmerkmale

Das im Embedded-Bereich teilweise noch weit verbreitete „not invented here“-Syndrom muss mehr und mehr durch eine „only invent things that make you unique“-Mentalität abgelöst werden.

Bediener und Bediengeräte anpassen können. Im Grunde steht heute noch in den Sternen, welche Art von Bedienungsterminals über welche Art von Kommunikationsmedium verfügbar sein wird. Mobiltelefon, Handheld-PC, PC, Autocomputer in allen möglichen Varianten über Telefonkabel, Breitbandkabel, Funk und Satellit sind denkbar, vieles wird noch neu hinzukommen. Nur wer in der Lage ist, Lösungsarchitekturen um sein Kerngeschäft zu konstruieren, welche die notwendige Offenheit und Flexibilität zur schnellen Integration bieten, kann seine Stärken mit den schnellen Fortschritten eines „Universellen Netzes“ kombinieren.

Konsequent weitergedacht führt diese Entwicklung zu immer weiteren Abstraktionen in der Softwareentwicklung, die Lösungen unterstützen, welche sich selbst an die konkrete Nutzungssituation (Ort, Zeit, Gerät, Nutzerprofil) anpassen. So ist es denkbar, dass sich Web-Pads selbst um das Bereitstellen aller Informationen und Funktionen kümmern, die der Nutzer aufgrund seiner Interessen, seiner derzeitigen Aufgabe, der Tageszeit und seines Standorts benötigt. Diese Dienste werden vor allem im „Universellen Netz“ erbracht und das

Dienstleistungsterminal ist dann lediglich für die Anforderung und Darstellung zuständig. Die Forschung agentenbasierender Methoden zeigt hier interessante Wege auf.



Quelle: ITEA 2001

Trend zu agentenbasierenden Softwarearchitekturen, die sich dynamisch selbst organisieren.

Prozesse und Toolketten

Doch Abstraktion besitzt gerade im Embedded-Bereich auch Nachteile:

- Die Verbindung zu den physikalischen Randbedingungen geht teilweise verloren.

-
- Zwischenschichten, die das Modell an das Betriebssystem oder die physikalische Plattform ankoppeln, kosten Speicher und Prozessorleistung.
 - Die nur teilweise mögliche Einbeziehung der physikalischen Realität in das Modell, wie beispielsweise des tatsächlichen zeitlichen Verhaltens, macht Iterationsschritte für das Verifizieren des Zusammenspiels nötig.

Hier kommt ein weiterer Aspekt zum Tragen: Die Entwicklungswerkzeuge und Dienstleistungen im Umfeld müssen zum einen solche Prozesse und zum anderen die schnelle Integration sowie das effektive Testen von Lösungen unterstützen. Gangbare Wege zeigen prozessunterstützende Technologien wie CASE-Tools in Kombination mit Prozessen wie ROPES (Rapid Object-oriented Process for Embedded Systems) oder RUP (Rational Unified Process).

Abgestimmte, durchgängige Toolketten, die sich in Prozesse eingliedern und eine weitgehend automatisierte Umsetzung von Modellen in physikalische Lösungen ermöglichen, werden die wichtigsten Impulse setzen. Und es wird sich auch an dieser Stelle zeigen, wer als Anbieter seine Kernkompetenzen in einer effektiven Arbeitsteilung mit den Kompetenzen von Partnern kombinieren kann. Nur so bekommt der Nutzer das, was er wirklich braucht: die durchgängige Unterstützung seiner Lösungsprozesse. Der Konsens ist also entscheidend. Der Embedded-Markt hat dazu noch wenig brauchbare Konzepte zu bieten, erste Ansätze sind jedoch erkennbar.

Prozessorleistung, Speicherbedarf, Bandbreite

Natürlich werden die oben beschriebenen Entwicklungen dazu führen, dass mehr Speicherkapazität und Rechenleistung erforderlich werden. Die Integration in ein „Universelles Netz“ erlaubt es, Aufgaben auf einen Server im Web auszulagern. Gleichzeitig wird sich die Entwicklung von Speicher- und Prozessortechnologien im bisherigen Tempo fortsetzen. Eher haben wir mit mageren Bandbreiten zu kämpfen, aber auch hier ist Land in Sicht, denn bis 2005 sollen sie medienübergreifend auf das Fünffache angewachsen sein. Außerdem ist mit noch ausgefeilteren Datenkompressionsverfahren zu rechnen.

Die technologische Basis für wirklich flexible und komplexe Lösungsarchitekturen ist also bereits in Ansätzen geschaffen und wird in den nächsten Jahren schrittweise ausgebaut. Der Embedded-Markt sollte sich deshalb intensiv mit den Möglichkeiten und Herausforderungen des „Universellen Netzes“ auseinandersetzen. Je schneller, desto besser für Anbieter wie Kunden. ■

Links zum Thema:

www.itea-office.org/documents/roadmap/itea_roadmap_download.html

www.iid.de/it-pro/kap4_2_1h.htm

www.iid.de/it-pro/kap4_3_1h.htm

Perspektiven

Die Zukunft im Embedded-Markt wird in vielen Fällen durch einen Paradigmenwechsel bestimmt:

- Viele Embedded-Systeme werden sich mehr und mehr zum Dienstleistungsterminal entwickeln, das dem Kunden durch eine stark personalisierte Kommunikation mit dem Internet der Zukunft („Universelles Netz“) wesentliche Zusatzleistungen bietet.
- Die Komplexität der Lösungen zwingt zu höherer Abstraktion in Form von standardisierten Prozessen und Modellierungsmethoden.
- Alle im Markt agierenden Anbieter werden neue Formen der Arbeitsteilung entwickeln müssen.
- Die Konsensfähigkeit zur Unterstützung eines sicheren „Universellen Netzes“, von geeigneten Entwicklungsprozessen bis hin zu durchgängigen Toolketten und Services wird ein wesentlicher Wettbewerbsfaktor sein.

Microsoft .NET: Einsatz in der Industrie

Egal ob Visual Basic, MFC und ATL oder die Internet-technologie ASP – alle diese Methoden zur Programmierung von Windows-Applikationen basieren auf dem Win32 API. Dessen Ende läutet Microsoft nun mit dem .NET Framework ein. Doch was bringt die neue Technologie dem industriellen Entwickler in der Praxis?

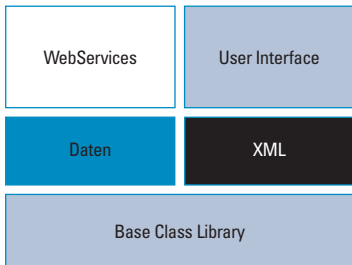
Wer die Einsatzmöglichkeiten von .NET im industriellen Umfeld bewerten möchte, muss sich zuerst generell mit der Grundidee der von Microsoft effektiv in Szene gesetzten Zukunftsstrategie vertraut machen.

So funktioniert .NET

Das Prinzip ist einfach: Das .NET Framework ist nichts anderes als eine Klassenbibliothek mit WebServices, User Interface, XML, Datenbankprogrammierung und Basisklassentechnologie. Ihren Kern bildet die Common Language Runtime (CLR), die alle Anwendungen unabhängig von der Sprache lauffähig macht, mit der sie programmiert sind. So gilt die CLR als Weiterentwicklung des bisherigen Integrationsmodells der COM-Technologie mit echten Mehrwerten:

- ein einheitliches und sprachübergreifendes Typsystem. Die Typen sind nicht mehr Bestandteil der Programmiersprachen, sondern des übergreifenden Frameworks;

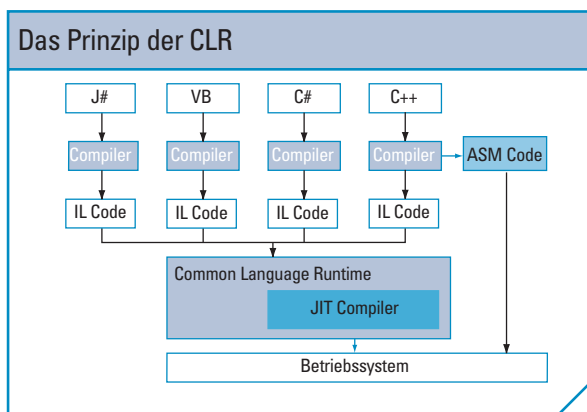
Aufbau von .NET



Die zentralen Bestandteile des .NET Frameworks

- ein einheitliches und erweiterbares Format für Metadaten, also Informationen, welche die Laufzeitumgebung braucht, um eine Applikation konfigurieren und Komponenten integrieren zu können;
- eine einheitliche und sprachübergreifende Laufzeitumgebung sowie
- eine einheitliche und sprachübergreifende Klassenbibliothek über alle Programmiersprachen hinweg. Wer sie beherrscht, kann in jeder beliebigen Sprache entwickeln. Der Unterschied liegt nur in der Syntax der Programmiersprache.

Ein wesentlicher Bestandteil der CLR ist der Just-in-Time (JIT) Compiler, eine mit Java vergleichbare Technologie. Programmiersprachen wie J#, Visual Basic, C# und C++ produzieren aus ihren Compilern heraus eine Intermediate Language (IL), die der JIT Compiler zur Laufzeit modulweise in Assembler- oder Maschinen-Code übersetzt. Dieses Verfahren ähnelt dem von Java mit dem Unterschied, dass



Der C++-Compiler kann als einziger auch direkt Assembler- bzw. Maschinen-Code als unmanaged Code erzeugen.

das Compilat in einer Art Cache vorgehalten wird und damit für weitere Aufrufe immer zur Verfügung steht. Wenn der IL-Code vom JIT Compiler interpretiert wird und dann in der CLR abläuft, spricht man von sogenanntem „managed Code“, weil die CLR

klare Restriktionen für das Laufzeitverhalten vorgibt. Das betrifft die Fehlerprüfung, Speicherverwaltung und Versionsprüfung benutzter Komponenten. Selbst Sprachen wie C++ stehen also nicht mehr im Maschinen-Code zur Verfügung. Wird aus dem C++-Compiler direkt Assembler- bzw. Maschinen-Code erzeugt, spricht man von „unmanaged Code“, der dann die CLR nicht nutzt und auf dem klassischen Win32 API basiert. C++ ist dabei die einzige Sprache, die heute (in der Beta 2) unmanaged Code erzeugen kann.

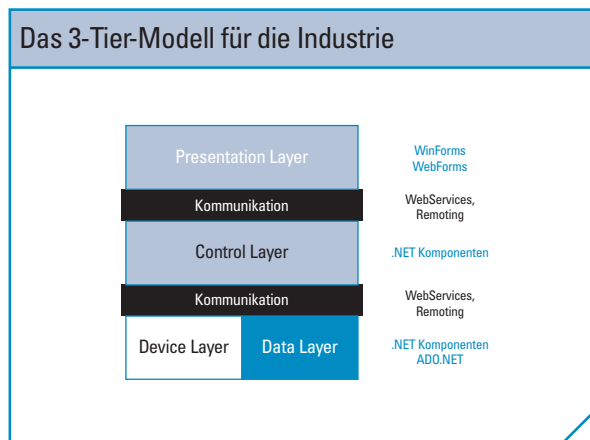
.NET in der Industrie

Die Anforderungen der Industrie sind klar umrissen. Es geht darum,

- Prozessdaten zu visualisieren,
- Feldbusse zu unterstützen,
- Software zu verteilen,
- Realtime-Fähigkeit zu unterstützen und
- speziell im Embedded-Bereich limitierte Ressourcen zu berücksichtigen.

Die .NET-Strategie von Microsoft ist hier genau zu beleuchten. So ist die Echtzeit-Fähigkeit mit virtuellen Maschinen, wie sie die CLR zur Verfügung stellt, aufgrund des JIT Compilers und der Garbage Collec-

tion kaum zu realisieren. Ein Blick auf das klassische 3-Tier-Modell soll zeigen, wo .NET heute industrielle Anwendungen konkret unterstützen kann.



Das .NET Framework unterstützt alle Layer für industrielle Anwendungen.

Das .NET Framework findet sich auf allen Ebenen des 3-Tier-Modells für den industriellen Einsatz wieder. Auf der obersten Schicht, dem Presentation Layer, werden die Prozessdaten mit WinForms und WebForms visualisiert. Da dieser Layer an beliebiger Stelle im Netzwerk ausgeführt werden kann, muss die Kommunikation zum Rest der Applikation sichergestellt sein. Hierfür bietet das .NET Framework die WebServices und das Remoting. Auf dem

darunter liegenden Control Layer sorgen die .NET Komponenten für die Steuerung und Regelung sowie die Abbildung der Prozesse. Auch hier realisieren die WebServices und das Remoting die Kommunikation – diesmal mit dem Device und Data Layer. Der Device Layer übernimmt mit weiteren .NET Komponenten die Abstraktion und Anbindung von externen Geräten und Systemkomponenten. Falls erforderlich, ermöglicht die ADO.NET-Technologie den Zugriff auf Datenbanken. In der Praxis sind dies häufig SQL-Server im Automatisierungsumfeld.

Wenn das .NET Framework scheinbar alle Layer abdecken kann, warum dann noch C, C++ oder Assembler? Einige Gegenfragen liefern die Antwort:

- Wie sollen in Zukunft Treiber erstellt werden? Langfristig werden sicher keine virtuellen Maschinen für Device-Treiber eines Betriebssystems auf den Markt kommen.
- Wie lässt sich die Echtzeit-Fähigkeit von WindowsCE.NET garantieren? Die Garbage Collection steckt der CLR hier unüberwindbare Grenzen.
- Wie sollen in Zukunft über 90 Prozent der Anwendungen, die nun einmal in C und C++ geschrieben sind, in den industriellen Bereich integriert werden? Sie müssten alle in moderne Programmiersprachen wie C#, J#, Visual Basic oder Managed C++ umgeschrieben werden.

Letztendlich werden auch zukünftig ältere und neuere Sprachen friedlich koexistieren, denn jede hat ihre ganz speziellen Einsatzgebiete und Qualitäten (siehe Tabelle rechts).

Fazit

Prinzipiell lohnt sich der Einsatz des .NET Frameworks umso mehr, je weiter oben der Entwickler im 3-Tier-Modell ansetzt. Nach unten hin kommt den klassischen Programmiersprachen wie C und C++ eine tragende Rolle zu. Schließlich spielt sich beispielsweise die Echtzeitfähigkeit vorwiegend im Device Layer, aber auch im Control Layer ab. Damit haben Unternehmen, die seit vielen Jahren mit C oder C++ arbeiten, ihre Investitionen langfristig gesichert. Doch der intensivere Blick auf .NET lohnt sich allemal, denn die Technologie zeigt dort echte Qualitäten, wo es um Visualisierung, ein Höchstmass an Internet Connectivity und verteilte Systeme geht. Und auch wenn C# erste Wahl für .NET ist, lassen sich C, C++ und Assembler dennoch integrieren. ■

C/C++	Die Programmiersprache der Wahl für Treiber und Realtime-Anwendungen unter CE
Managed C++	„Neue“ Programmiersprache, ohne Pointer, mit Erweiterungen für die Unterstützung der CLR. Gut geeignet für die Integration von altem C/C++-Code
C#	Zur Zeit optimale Programmiersprache für die Programmierung der CLR
Visual Basic	Im industriellen Bereich zur Zeit wenig verbreitet. .NET hat die Laufzeitumgebung durch das .NET Framework stark verändert.
J#	Entwickelt für den optimalen Umstieg von J++

.NET unterstützt die Industrie

Folgende Bestandteile des .NET Frameworks sind für den industriellen Einsatz prädestiniert:

WebForms

- Ersetzen die MFC- und Visual Basic-Programmierung
- Volle Rapid Application Development (RAD)-Unterstützung (damit auch für Prototyping geeignet)
- Erlauben es, extrem leistungsfähige formularbasierte Internetanwendungen zu erstellen.
- Verwendung „echter“ Programmiersprachen
- Seite wird beim ersten Aufruf kompiliert und als HTML an Browser gesendet
- Verwendung von HTML Controls (rückwärtskompatibel) und von Web Controls

WinForms

- Volle Internet-Fähigkeit für den Presentation Layer
- Setzt völlig neue Maßstäbe für den Bau von Applikationen, für Fernwartung, -diagnose und -steuerung
- Inklusive GDI+ für ansprechende Grafiken

.NET Komponentenmodell

- Ersetzt den Inprocess Server oder das bisherige DLL-Modell
- Liefert Assemblies als DLLs
- Shared Assemblies unterstützen die Versionierung
- Sehr einfach im Deployment:
 - XCopy-Deployment
 - Keine Registry-Einträge mehr nötig
 - Shared Assemblies werden in den Global Assembly-Cache eingetragen.
- Einfaches Konzept für die Modularisierung im Prozess

Remoting

- Ermöglicht eine einfache Verteilung von Komponenten im Netz
- Ersetzt DCOM
- Einfach in der Programmierung

- Einfach in Handhabung und Installation
- Schließt das Transaktionsmonitoring ein (interessant nur beim Einsatz von Datenbanken)
- Remoting erfordert keine End-Point- zu End-Point-Kommunikation (Abbildung beliebiger Protokolle aus Sicht von .NET, beispielsweise Feldbusse)
- Implementierung des Control und Device Layers

WebServices

- Bilden Funktionen (Methoden) über das Simple Object Access Protocol (SOAP) im Internet ab.
- Ideal für Fernwartung, -diagnose und -steuerung sowie die Abbildung von Prozessen
- Kommunikation zum
 - Control Layer (Abbildung von Prozessen im Internet)
 - Device Layer (Fernwartung und -diagnose von Systemkomponenten und externen Geräten)

XML

- Ersetzt die Konfigurationsdateien und die Registry
- Beliebiger Zugriff auf die Daten über Xpath
- Einfache Konvertierung durch XSLT
- Programmierung der XML-Dateien im .NET durch DOM und SAX
- Bildet die Datenhaltung für kleine Datenmengen und Konfigurationsdaten ab

InterOp

- ermöglicht:
 - Integration bestehender Applikationen
 - Weiche Migration bestehender Sourcen ins .NET
- Techniken:
 - Aufruf von Funktionen in DLLs
 - Einbinden von COM-Komponenten
 - Einbinden von ActiveX-Controls
 - Kombination von Managed und Unmanaged C++ als Gateway

ADO.NET

- Zur Anbindung von Datenbanken, falls erforderlich

Kurse zum Thema

.NET mit C# (.NET/C#)

Teilnehmer:.....Alle, die mit dem neuen .NET Framework von Microsoft Applikationen erstellen wollen

Vorkenntnisse:..Fundierte Grundkenntnisse in der Windows-Programmierung mit MFC, Win32 oder Visual Basic

Kursziel:.....Applikationen mit C# effizient programmieren; Benutzen aller grundlegenden Programmier-elemente von C#; optimales Anwenden des .NET Frameworks für WinForms; WebForms und Web Services; praktische Tipps für das Anbinden von C- oder C++-Code

Kursinhalt:.....Einführung in .NET-Strategie, Common Language Runtime (CLR), Assemblies und deren Versionsverwaltung; C#; WinForms; Webforms; Web Services; Remoting; Migration und Einbindung von C++-Applikationen in C#

Termine:.....18.03.-22.03.2002; spätere Termine unter www.microconsult.de

Preis:.....1.740 Euro + Ust.

Neue .NET-Seminare

Der Kurs .NET mit C# wird im Herbst 2002 in drei Einzelseminare aufgliedert, um die unterschiedlichen Anforderungen der Teilnehmer gezielt abzudecken:

■ Programmieren mit C# (C#/PRG)

Neben einer Einführung in .NET erfahren die Teilnehmer alles Wissenswerte über C#, Ereignisse und Delegates, Assemblies sowie WinForms.

■ Verteilte Systeme mit .NET (.NET/VO)

Nach einer Einführung zum Thema „verteilte Systeme“ widmet sich der Kurs dem Remoting, SOAP, ASP.NET, WebServices, Security, Deployment sowie ADO.NET.

■ Programmieren mit WinForms (.NET/FORM)

Hier dreht sich alles um WinForms, Datei, Multitasking, Interop, ASP.Net, ADO.NET und Deployment.

Einführung in die .NET Programmierung mit Visual Basic (.NET/VB)

Teilnehmer:.....Alle, die mit Visual Basic programmieren wollen

Vorkenntnisse:..Kenntnis einer Hochsprache

Kursziel:.....Applikationen mit VB.NET; Anwendung grundlegender Programmier-elemente; Schaffung wesentlicher Voraussetzungen für das Programmieren des .NET Frameworks

Kursinhalt:.....Merkmale und Konzepte von VB.NET; die BASIC-Sprache von VB.NET mit Datentypen, Operatoren und Kontrollstrukturen; Erstellen von Benutzeroberflächen; Programmier-technik

Termine:.....11.02.-15.02.2002; spätere Termine unter www.microconsult.de

Preis:.....1.690 Euro + Ust.

OOA Analyse mit der UML (OOA/UML)

Teilnehmer:.....Projektleiter in der Software-Entwicklung, Anwendungsentwickler, OSW-Entwickler im Embedded Bereich

Vorkenntnisse:..Programmiererfahrung (CHILL, C, Fortran, C++ und andere)

Kursziel:.....Kompetenter Einsatz von Analyse, Entwurfsverfahren und Darstellungsform der Unified Modeling Language (UML)

Kursinhalt:.....OOP; objektorientierte Basiskonzepte und deren Darstellung mit Hilfe der UML Klassennotation; dynamisches Verhalten objektorientierter Software; Umsetzung in verschiedene Programmiersprachen

Termine:.....04.03.-07.03.; 08.04.-11.04.2002; spätere Termine unter www.microconsult.de

Preis:.....1.430 Euro + Ust.

UML AufbauSeminar (OOA/UML-MT)

- Teilnehmer:**.....Projektleiter in der Software-Entwicklung
- Vorkenntnisse:**..Grundwissen aus dem MicroConsult-Seminar „OOA Analyse mit der UML“
- Kursziel:**.....Verschiedene Vorgehensmodelle richtig einschätzen; optimaler Einsatz von Techniken zur Modularisierung; erfolgreiche Planung verteilter und paralleler Systeme
- Kursinhalt:**.....Software-Analyse (Unified Modeling Process, MSF); UML-Schichtenmodelle; Software-Verteilung; Parallelität
- Termine:**.....15.04.-19.04.2002; spätere Termine unter www.microconsult.de
- Preis:**.....1.790 Euro + Ust.

Objektorientierte Analyse für Embedded Systeme (OOA/UML-RT)

- Teilnehmer:**.....Software-Entwickler von Embedded Systemen
- Vorkenntnisse:**..Projekterfahrung mit Embedded Systemen und Grundwissen aus dem MicroConsult-Seminar „OOA Analyse mit der UML“
- Kursziel:**.....Optimaler Einsatz der UML; professionelle Nutzung von Frameworks; effiziente Entwicklung von Architekturen in Realtime-Systemen
- Kursinhalt:**.....UML und Embedded Systeme; Vorgehensmodelle; grundlegende Software-Architekturen; Modellierung von Echtzeitsystemen; Internetarchitektur
- Termine:**.....25.02.-01.03.; 22.04.-26.04.2002; spätere Termine unter www.microconsult.de
- Preis:**.....1.790 Euro + Ust.

Software-Qualität für Embedded Systeme (SW/QS)

Teilnehmer:.....Software-Entwickler von Embedded Systemen, für Software-Qualität verantwortliche Entwickler und Manager, Qualitätsbeauftragte

Vorkenntnisse:..Projekterfahrung mit Software-Systemen

Kursziel:.....Systematisches Vorgehen der Software-Qualität, Kenntnis von diesbezüglichen Standards, Bestätigung der erreichten Qualität

Kursinhalt:.....Qualität von Entwicklungsprozessen und der Produkte (Hardware und Software); Bestätigung der Produktqualität; praktische Übungen zum Ableiten von nicht funktionalen Anforderungen aus Standards, zum Prüfen von Code (C51, C167) und zum Bewerten von Prüfergebnissen

Termine:.....04.02.-08.02.2002; spätere Termine unter www.microconsult.de

Preis:.....1.790 Euro + Ust.

Intra-/Internet-Anbindung von Embedded Systemen (EMB-NET)

Teilnehmer:.....Hard- und Software-Entwickler, Projektleiter

Vorkenntnisse:..Grundkenntnisse über die Architektur von Mikrocontroller-Systemen, Programmiererfahrung in C

Kursziel:.....Embedded Systeme an das Intra-/Internet anbinden, Embedded Webserver implementieren, Intra-/Internet-Seitenprogrammierung.

Kursinhalt:.....Grundlagen Intra-/Internet-Technologie; TCP/IP; HTML- und JavaScript-Seitenprogrammierung; Erstellen und Einbinden von Java Applets; Intra-/Internetanbindung von Embedded und Deeply Embedded Systemen mit emWare; Grundlagen und Implementierung eines Embedded Webserver basierend auf einem Echtzeitbetriebssystem

Termine:.....18.03.-22.03.2002; spätere Termine unter www.microconsult.de

Preis:.....1.710 Euro + Ust.

Workshop Internetprogrammierung (INET/PRG)

Teilnehmer:.....Entwickler, Projektleiter, Webadministratoren und
-autoren

Vorkenntnisse:..allgemeine Programmierkenntnisse

Kursziel:.....fundierte Kenntnisse zum Erstellen von Webseiten
mit HTML und JavaScript, Überblick über aktuelle
Internettechnologien, ASP, CGI und PHP

Kursinhalt:.....Einführung in die Internet-Begriffswelt; Architektur
einer Webapplikation; HTML; JavaScript als client-
seitige Programmiersprache; Überblick über ser-
verseitige Programmierung; XML und XHTML

Termine:.....22.04.-26.04.2002; spätere Termine unter
www.microconsult.de

Preis:.....1.660 Euro + Ust.



Weitere Informationen

Andreas Munk – Ihr Vertriebskontakt

Tel.: +49 (0) 89 - 45 06 17 - 42

E-Mail: a.munk@microconsult.de



MICRO CONSULT

MicroConsult GmbH • Schule für MicroElektronik & Informationstechnologie
Rosenheimer Straße 143 b • D-81671 München • Tel.: (089) 45 06 17- 71
Fax: (089) 45 06 17- 17 • www.microconsult.de