

Vortrag auf der SPS/IPC/DRIVES 2008 in Nürnberg

Web-HMI in der Praxis – Ein Erfahrungsbericht zu innovativen Visualisierungstechnologien im Maschinen- und Anlagenbau

Joern Kowalewski, macio GmbH

Rita Marnau, macio GmbH

Sven Schreier, macio GmbH

Web HMIs in industry applications - a hands-on report

A concept of machine / plant supervision systems is presented. Due to safety concerns, the „control“ part of „SCADA“ is a classical desktop application, but the supervisory functions have been implemented as a set of browser-based web applications around JSF and Ajax technologies. Missing pieces in an open-source-IDE tool chain so edit such HMIs graphically are addressed.

SCADA, Ajax, Web HMI, JSF

1. Einleitung

1.1 Ortsbestimmung Anlagensteuerung vs. Anlagenüberwachung

Tief in Anlagen integrierte SCADA Systeme übernehmen neben der Überwachung von Maschinen und der Sammlung von Produktions- und Prozessdaten vielfach auch steuernde oder zumindest den Prozess beeinflussende Funktionen. Gerade bei Anwendungen, bei der durch Fehlbedienung erhebliches Gefahrenpotential besteht, wie zum Beispiel die Steuerung von Maschinen, ist es unumgänglich, dass eine Steuerungsfunktion nur von dem dafür berechtigten Personal ausgeführt werden kann und das auch nur von einem Ort aus, wo die komplette Einsicht der relevanten Maschinen- bzw. Anlagenfunktionen gewährleistet ist.

Für derartige Anwendungen empfiehlt sich der Einsatz von nach dedizierten Kriterien verteilten Desktop-Applikationen. Die Installation und Konfiguration des einzelnen HMI-Frontends an jedem Arbeitsplatz hat zwar einen deutlichen Mehraufwand bei der Einrichtung der Anlage zur Folge, jedoch können die Sicherheitsanforderungen bezüglich der Fragen: „Wer darf welche Funktionen an welchem Arbeitsplatz bedienen?“ durch solche „vollen“ JAVA oder C# Applikationen recht gradlinig erfüllt werden.

Ein anderes Thema ist die reine Überwachung von Maschinen und Anlagen. Wenn es nur um die Anzeige von Daten geht, ist die Ortsabhängigkeit kein kritischer Faktor mehr. Um eine Übersicht über den aktuellen Zustand einer Produktionsstrecke zu erhalten, bieten sich somit Webapplikationen an. Diese können von so genannten Thin Clients ohne weitere Vorinstallation an jedem Ort mit Netzzugang angesprochen werden. Beim Auftreten von Fehlern oder Abweichungen können dann die notwendigen Personen vor Ort benachrichtigt und die entsprechenden Maßnahmen eingeleitet werden. Diese Webapplikationen sollen auch auf gängigen PDAs oder Handys laufen können, und sie müssen robust über Firewalls hinweg funktionieren.

Die von uns verfolgten Konzepte sehen also eine strikte Trennung der Applikationsarten nach reinen Desktop Lösungen für die sichere Steuerung einer Anlage und rein Browser basierten Lösungen für deren Überwachungs- und Informationssysteme vor. Letztere sollen hier näher betrachtet werden.

1.2 Projektierung

Gerade im Anlagenbau ist die HMI Erstellung weniger die Aufgabe von Informatikern als die von Prozess-Spezialisten, deren KnowHow in den zu modellierenden Abläufen und nicht in der objektorientierten Programmierung liegt. Im Desktop Bereich lassen dafür mit grafischen Editoren und einem „Variablenbrowser“ für den Prozessdatenpool z.B. als Eclipse Erweiterung mit vertretbarem Aufwand „Drag & Drop“- Werkzeuge herstellen.

Damit können aus vorgefertigten GUI-Elementen brauchbare Bedienscreens generiert werden. Eine vergleichbare Toolchain für die hier näher untersuchten Technologien existiert bisher (bei uns) noch nicht auch darüber wird hier berichtet.

2. Anforderungen an eine Webapplikation

Die Webapplikation soll unabhängig vom aktuellen Aufenthaltsort des Nutzers Zustandsdaten publizieren sowie Datenverdichtungen und Auswertungen generieren. Der Benutzer muss dabei interaktiv in komplexen Datenmengen oder Zusammenhängen navigieren können, Korrelationen vorgeben und Ergebnisse zusammenstellen. Außerdem soll sie jederzeit über den aktuellen Zustand von wichtigen Maschinenparametern informieren. Eine zeitnahe, automatische Aktualisierung von Werten ist somit eine Grundvoraussetzung. Kurz, eine einfache Webseite, die auf Anforderung mehr oder minder statische Daten anzeigt, genügt an dieser Stelle nicht. Im Grunde werden alle interaktiven Vorteile einer echten Desktop Applikation erwartet. Demnach sind folgende Anforderungen bei der Realisierung einer solchen Webapplikation zu beachten:

- Lokale Verfügbarkeit der Komponenten
- Sicherheit
- Funktionalität
- Benutzbarkeit, Attraktivität

2.1 Lokale Verfügbarkeit der Komponenten

Die Webapplikation muss unabhängig von speziellen Browsern oder der Verwendung von Plugins oder proprietären Runtime Erweiterungen sein. Dies ist wichtig, um eine universelle Ortsunabhängigkeit zu erreichen. Das heißt, die Webapplikation soll immer überall laufen, ohne dass vorher eine Installation oder Konfiguration notwendig wäre. Auch Deployment Probleme kann man auf diese Weise einfach umgehen. Des Weiteren sind auf heutigen Hand Held Devices / Handies proprietäre Plugins in der Regel nicht verfügbar. Dieser Anspruch an Unabhängigkeit schließt die Verwendung von vielen interessanten Technologien wie „Flex/Flash/Silverlight“ aus. Wie wir im Folgenden zeigen, gibt es Technologien, die allen oben genannten Ansprüchen gerecht werden und ohne zusätzliche Plugins oder Lizenzen auskommen.

2.2 Sicherheit

Eine andere Einschränkung betrifft die Rechteverteilung, denn Zustandsdaten der Maschinen, Aufträge etc. dürfen nicht für alle und jeden zugänglich sein. Eine Webapplikation mit eingeschränkten Sichtbereichen, die durch eine sichere Authentifizierung den Nutzern bestimmte Rechte und damit den Zugriff auf unterschiedliche Daten einräumt, ist hier gefordert. Als Kommunikationsprotokolle sind ausschließlich verschlüsselte Varianten wie https: zu verwenden. Besondere Ports sollten allerdings nicht verwendet werden. Deren Nutzung scheitert meist an der ersten im Unternehmen installierten Firewall. Ein Authentifizierungsmechanismus kann ähnlich wie bei Desktopapplikationen auch bei Webapplikationen durch Passwort geschützte Bereiche und Rollenverteilung mit unterschiedlichen Rechten erreicht werden.

2.3 Funktionalität

Soll eine Webapplikation zur Überwachung von Maschinen eingesetzt werden, sollte sie auch genau diese Aufgabe erfüllen. Dies bedeutet:

- permanent aktuelle Daten
- die wichtigsten Daten auf einen Blick
- Fehler und Warnungen unübersehbar und eindeutig
- Eindeutige Zuordnung der Daten zum Ursprung
- Statistiken und evtl. Vorausberechnungen
- Einfache und komfortable Navigation in Datenbeständen

Es ist unerlässlich, dass die angezeigten Daten permanent aktuell sind, so dass sich der Anwender auf die Daten verlassen kann. Dies muss ohne Eingreifen des Anwenders gewährleistet sein. Die Daten sollten sich automatisch periodisch aktualisieren. Diese Periode kann je nach Anwendung im Millisekunden- (nur im LAN) oder Sekundenbereich sein. Durch den Einsatz neuester Webtechnologien ist dies realisierbar. Die Präsentation der Daten ist eine Frage des Designs.

2.4 Benutzbarkeit, Attraktivität

In industriellen Applikationen ist das Schlagwort „Usability“ längst kein Fremdwort mehr. Auf einen Blick sollte der Anwender erfassen können, was ihn am meisten interessiert. Wichtig ist auch, eine eindeutige Zugehörigkeit der Parameter zu gewährleisten. Es muss immer klar sein, aus welcher Quelle Daten oder

Fehlermeldungen stammen. Und letztlich zeichnet sich jedes gute HMI durch eine klare und sichere Navigation aus. Gerade auf Desktoprechnern werden heute hervorragend bedienbare Programme geschrieben. Durch den Einsatz von flexiblen Komponentenbibliotheken oder dem Entwurf eines eigenen Look and Feel, kann z.B. mit Hilfe von Java professionelles Layout und intuitive Bedienbarkeit mit einfachen Mitteln erreicht werden.

Diesen Standard haben Webapplikationen heutzutage noch nicht erreicht. Der Anspruch an eine gute Usability gilt dennoch unverändert. Gerade große Datenmengen müssen graphisch so aufbereitet werden, dass Ergebnisse oder Trends leicht erfassbar sind. Applikationen deren Fenster nicht im Zentrum der Aufmerksamkeit des Betrachters liegt, müssen wichtige Zustandsänderungen deutlich, aber ohne „visuellen Lärm“ kund tun. Die Geschwindigkeit, mit der Daten zur Verfügung stehen, ist ebenfalls ein Punkt, der bedacht werden sollte. Nichts ist lästiger, als „Minuten“ auf das Neuladen einer Tabelle mit 2000 Einträgen warten zu müssen, nur weil man ein Filter neu setzt.

3. Technologien

Jede „Web Entwicklung“ sollte sich schon aus Aufwandsgründen aber auch wegen der im Industriebereich erwarteten Langlebigkeit von Standards eng an bereits breit etablierte Technologien halten. Im Folgenden werden unsere Erfahrungen mit einer Auswahl dieser Technologien erläutert.

3.1 http/HTML

Es gibt unterschiedliche Möglichkeiten, eine Webapplikation zu realisieren. Die Berechnung der Daten kann entweder auf dem Server oder auf dem Client erfolgen.

Die einfachste klassische Realisierung ist eine Request-Response Kommunikation per http zwischen Client und Server. Hierbei werden immer komplette HTML Seiten verschickt. Die Berechnung der Daten erfolgt auf dem Server. Eine Veränderung der Daten kann allerdings nur auf dem Browser angezeigt werden, wenn der Client danach gefragt hat und die komplette Seite aktualisiert wird.

Die Gestaltungsmöglichkeiten sind bei einer reinen HTML Seite beschränkt. Mit Hilfe von CSS (Cascading Style Sheets) ist es möglich, sich Steuerelemente und eigene Komponenten zu kreieren. Allerdings ist dies etwas aufwendig und setzt Erfahrung in dem Bereich voraus. Ein automatisches Update der Daten ist mit der alleinigen Verwendung von HTML und http nicht möglich.

3.2 Java Applets

Java Applets erfreuen sich großer Beliebtheit und werden sehr häufig für Webapplikationen eingesetzt. Die Berechnung der Daten findet hier auf der Client Seite statt. Java Applets ermöglichen im Gegensatz zur Request-Response Kommunikation von http eine bidirektionale Verbindung mit dem Server. Dies bedeutet, dass der Server von sich aus Daten schicken kann, ohne dass der Client explizit danach gefragt hat.

Java Applets erfordern jedoch eine Java Runtime auf dem Zielsystem. Ein weiteres Problem bei Java Applets sind die Firewalls auf der Kommunikationsstrecke. Java Applets benötigen für die Datenübertragung einen zusätzlichen Port. Der Port „80“, welcher für den Internetverkehr der http Kommunikation reserviert wird, wird in der Regel gut kontrolliert. Die Öffnung weiterer Ports wird wegen der prinzipiell entstehenden Sicherheitslöcher von sehr vielen IT Abteilungen abgelehnt.

3.3 Java Script

Wir wollen hier eine klar Unterscheidung zwischen Java Script und Java Applets machen, da beide Technologien häufig in einem Atemzug genannt werden. Java Script ist lediglich die Methodik auf einer Webseite Interaktion des Benutzers zu ermöglichen, in dem in den HTML Code java-ähnliche Funktionen eingebunden werden. Java Applets sind eigenständige Applikationen, die entweder in die Webseite mit eingebunden werden, oder diese vollständig ersetzen. Java Script ist eine funktionale Sprache, die jeder moderne Browser mit zunehmend höherer Performance beherrscht. Die Java Script Funktionen, ermöglichen komfortable Interaktionen. Zum Beispiel kann die Anwendung eines Tabpane oder das Auftauchen eines Popup durch Javascript realisiert werden. Java Script ist die Grundlage für alle auf java-basierenden Frameworks und hat das Web 2.0 wie wir es heute kennen, erst ermöglicht.

3.4 Java Server Faces

Java Server Faces ist ein auf Java basierendes Framework für Webapplikationen, dass auf Komponenten basiert. Das Ziel von Java Server Faces ist es, eine klare Trennung, zwischen Datenmodell, Layout (View) und Berechnung der Daten (Controller) zu erreichen. Bei der bisherigen auf Java basierenden Lösungen, dynamisch Webseiten zu erzeugen, ist html und java code sehr ineinander verschachtelt gewesen. Erweiterungen oder Fehlersuche gestalteten sich so schwierig. Bei Java Server Faces erfolgt eine klare Trennung. JSF bietet eine vielseitige Komponentenbibliothek. Steuer- und Anzeigeelemente können somit bequem verwendet werden. Auch das Anpassen der Komponenten auf ein individuelles Farbschema, oder die Kreierung eigener Komponenten, ist mit JSF möglich. Für unsere Webapplikationen haben wir mehrfach Java Server Faces verwendet und damit sehr gute Erfahrungen gemacht.

3.5 Ajax

Eine klassische Web-Applikation mit einer zustandslosen HTTP-Verbindung benötigt immer eine normale HTTP-Anfrage vom Client an den Server, bevor der Server eine komplette neue Seite als Antwort ausliefert. Auch wenn sich nur ein einzelnes Datum auf der Seite geändert hat, muss immer die komplette Seite neu übertragen und geladen werden, Stockt die Datenlieferung durch einen internen Fehler oder z.B. Routingprobleme ist auch der Bedienfluss der Applikation aus Sicht des Anwenders unterbrochen. Der Begriff Ajax steht für Asynchronous JavaScript and XML und ist ein Konzept zur asynchronen Kommunikation zwischen Webserver und Browser.

Dies ermöglicht es, eine HTTP-Anfrage durchzuführen um z.B. Formulareingaben serverseitig zu verifizieren und das Ergebnis im Browser anzuzeigen, ohne die ganze HTML-Seite neu zu laden.

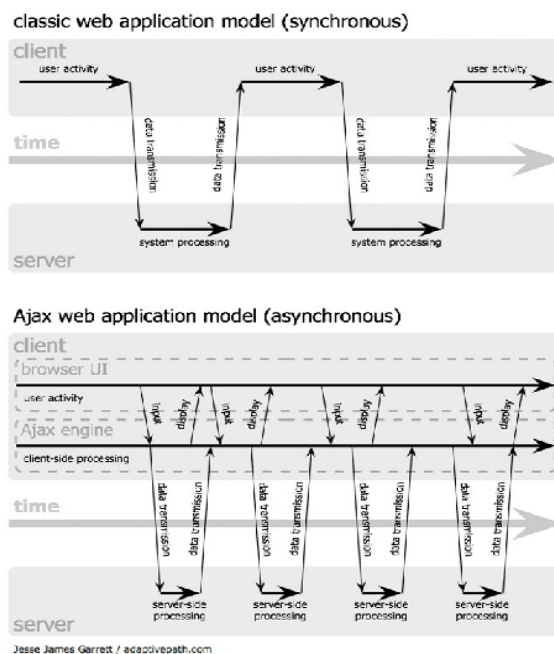


Abbildung 1: Prozessfluss einer klassischen Web-Applikation gegenüber einer Ajax-Applikation.

3.6 Reverse Ajax

Im Vergleich zu einer Desktop-Applikation gibt es bei Web- und auch bei Ajax-Applikationen einen wichtigen Unterschied: Es können keine Ereignisse vom Server zum Client geschickt werden, da das zustandslose HTTP-Protokoll immer zuerst eine Anfrage des Clients vorsieht. Um das Problem zu lösen, gibt es verschiedene technologische Ansätze, die oft unter dem Begriff „Reverse Ajax“ zusammengefasst werden:

- Polling: Der Client fragt zyklisch (z.B. alle drei Sekunden) beim Server an, ob neue Daten vorliegen. Bei vielen gleichzeitig auf Änderungen zu überwachenden Daten kann dies zu deutlich höherem Datenverkehr und ständiger Belastung von Client und Server führen.

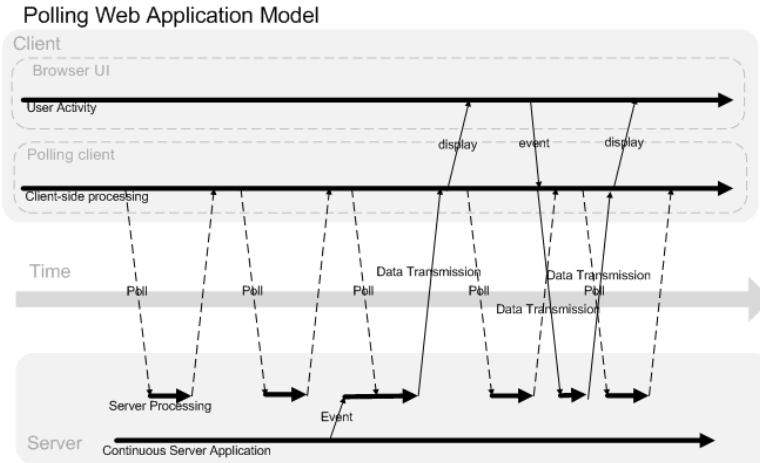


Abbildung 2: Prozessfluss Polling

- Comet: Die HTTP-Verbindung zwischen Client und Server wird so lange offen gehalten wie möglich. Dies wird erreicht, indem Dummy-Daten möglichst langsam ausgeliefert werden. Damit alle wichtigen Elemente der Seite trotzdem komplett geladen werden, wird ein Trick angewendet: Ein Iframe wird versteckt in die Seite integriert und nur der Inhalt dieses Iframes wird so langsam wie möglich ausgeliefert, der sichtbare Rest der Seite wird vorher komplett übertragen und angezeigt. Über diese lange offen gehaltene Verbindung kann der Server nun Daten an den Client schicken, auf Client-Seite werden sie per JavaScript verarbeitet.

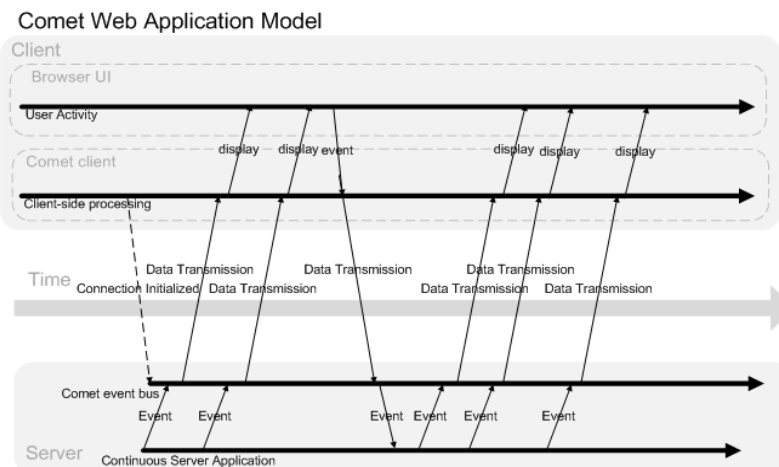


Abbildung 3: Prozessfluss Comet

- Piggyback: Daten, die vom Server an den Client geschickt werden sollen, werden gesammelt und bei der nächsten Anfrage des Clients „huckepack“ mit der Antwort auf die eigentliche Anfrage übertragen. Diese Technik eignet sich nur für Applikationen bei denen der User öfter neue Anfragen stellt und wirdentsprechend selten eingesetzt.

Bei Updatezyklen im Sekundenbereich haben wir eine geschickt angewendete Polling Strategie in unseren Projekten stets als ausreichend empfunden.

4. Projektierung

Das hier verfolgte Konzept sieht vor, Desktopapplikationen zur Steuerung von Maschinen zu verwenden, und Webapplikation zu deren (Fern)Überwachung. Beide Applikationsarten werden von ausgebildeten Softwareentwicklern (Informatikern etc.) entwickelt. Der Idealfall wäre, wenn die Erstellung oder zumindest die Erweiterung der entsprechenden Bedienoberflächen ohne wesentliche Programmierkenntnisse, direkt durch Prozess Spezialisten (z.B. Verfahrenstechniker) möglich wäre. Traditionelle Scada Systeme bieten zur Lösung dieses Problems ein oder mehrere Projektierungstools, mit welchen man per „Drag and Drop“ Komponenten in ein Fenster ziehen, mit einfachen Zeichenfunktionen zu Anlagenbildern verbinden und per Mausklick die entsprechenden Variablen zuordnen kann. Perfekt wäre es, wenn es durch ein und dasselbe Tool möglich wäre, eine Desktopapplikation zur Steuerung und eine verwandte Webapplikation zur Überwachung zu erstellen, oder gar die relevanten Teile der Desktopapplikation mit ein paar Mausklicks in eine Webapplikation umzuwandeln.

4.1 Projektierungstool für Desktopapplikationen

Ein Projektierungstool für Desktopapplikationen zu entwickeln, ist mit vertretbarem Aufwand realisierbar. Die Eclipse RCP Umgebung und der Visual Editor von Eclipse liefern die nötigen Voraussetzungen. Auf dieser Basis kann mit eigenen Erweiterungen ein individuelles Projektierungswerkzeug erstellt werden. Selbst die Bindung von Prozessvariablen an die einzelnen Komponenten ist per XML Konfiguration machbar. Somit kann auch ein Anwender ohne OO-Programmierkenntnisse, seine Applikation selbstständig erweitern kann.

4.2 Projektierungstool für Webapplikationen

Ein einziges Projektierungstool zu entwickeln, das sowohl für Desktopapplikationen als auch für Webapplikationen zu verwenden ist, ist ohne weiteres nicht möglich. Das Problem sind die sehr unterschiedlichen Darstellungskomponenten. So verwenden wir für den Desktop Swing und bei einer Webapplikation JSF Komponenten. Beide lassen sich nicht ineinander konvertieren. Für eine Webapplikation muss also ein eigenes Tool entwickelt werden. Die Wiederverwendung bezöge sich damit zwar lediglich auf das Design und die Arbeitsabläufe, welche bei eng verwandten Werkzeugen nur einmal erlernt werden müssten, wäre aber dennoch wirtschaftlich hinreichend interessant. In kundenspezifischen Projekten gilt auch hier wieder die Voraussetzung der breiten Verfügbarkeit der Basistechnologien. Unsere Planung sah darufhin zunächst vor, diese Werkzeuge wieder auf Basis von Eclipse herzustellen. Leider haben unsere Forschungen

ergeben, dass sich derzeit zur Entwicklung eines Projektierungstools für Webapplikationen am ehesten NetBeans als Plattform verwenden lässt. Als Technologien empfehlen sich Dynamic Faces oder ICEFaces. Beide sind NetBeans kompatibel und unterstützen Reverse Ajax. Allerdings lässt sowohl die Güte der Dokumentation als auch die Struktur der allgemein genutzten Schnittstellen viele Wünsche offen so dass wir derzeit über ein Funktionsmodell noch nicht hinaus gekommen sind.

5. Fazit und Ausblick

Für die Programmierung von hochdynamischen Web Oberflächen stehen eine Reihe gut eingeführter Technologien zur Verfügung. Mit JSF und Ajax / Reverse Ajax gelingt es ansprechende Browser-basierte HMIs zu bauen, die ohne zusätzliche Plugins arbeiten und ausschließlich mit Firewall-freundlichen Protokollen auskommen.

Die einfache Projektierung von Web und Desktop Applikationen mit einem einzigen, vorzugsweise auf Eclipse basierten Werkzeug darzustellen ist uns noch nicht gelungen. Für die Projektierung von Browser basierten HMIs in diesen Technologien haben wir derzeit keine öffentlich zugänglichen Entwicklungsprojekte gefunden, die weit genug fortgeschritten sind, um im Rahmen von einzelnen Kundenprojekten zu brauchbaren grafischen „Drag&Drop“ Editoren ausentwickelt werden zu können. Viel versprechende, neue Universalwerkzeuge wie Adobe Flex oder MS Silverlight kommen ihrerseits nicht ohne Browsererweiterung aus.

Zumindest in Bereichen mit Gefahrenpotential halten wir die strikte Trennung von Bedienapplikationen und Beobachtungsapplikationen nach wie vor für sinnvoll. Nur letztere können ohne zusätzliche Sicherungsmassnahmen Browser-basiert aufgebaut werden.

Quellenangaben

Bilder 1,2,3 : Sprocketonline (2006), Exploring Reverse AJAX (Stand 26.08.2008)
<http://gmapsdotnetcontrol.blogspot.com/2006/08/exploring-reverse-ajax-ajax.html>

Autorenadresse: macio GmbH, Am Kiel-Kanal 1, D-24106 Kiel --- www.macio.de

Teile des Inhalts wurden im Rahmen der Bachelors Thesis „Toolunterstützte Projektierung von webbasierten Benutzeroberflächen“ von Herrn Henning Kröger in Zusammenarbeit der macio GmbH und der Fachhochschule Kiel entwickelt.

Über die macio GmbH:

Die macio GmbH entwickelt und realisiert seit 2002 Software-Lösungen im Kundenauftrag für den Maschinen-, Geräte- und Anlagenbau.

Als kompetenter Partner unterstützt das Kieler Unternehmen seine Kunden bei der Realisierung maßgeschneiderter Softwarelösungen auf embedded und PC-Basis zum Bedienen, Beobachten und Steuern von Maschinen.

Einzigartig ist die inhouse Kombination von Design- und Ingenieur-Know-how bei der Konzeption von Human Machine Interfaces (HMI). macio bietet seinen Kunden Softwaretechnologie der Spitzenklasse, die das Profil und die Marke des Kunden in den Vordergrund stellt.

Die macio GmbH leistet so einen hohen Beitrag zur Qualität und Wertigkeit der Produkte ihrer Auftraggeber.

Kontakt Daten:

Alexander Friedel

macio GmbH

Am Kiel-Kanal 1

D - 24106 Kiel

Phone: 0431 67072-20

Fax: 0431 67072-29

Mobil: 0176 23383968

www.macio.de

alexander.friedel@macio.de