# XML-basierte Internetanbindung technischer Prozesse

Stephan Eberle

Institut für Automatisierungs- und Softwaretechnik (IAS)
Universität Stuttgart
Pfaffenwaldring 47, D-70550 Stuttgart
eberle@ias.uni-stuttgart.de

**Kurzfassung.** In zunehmendem Maße erlangt XML die Bedeutung einer Schlüsseltechnologie beim Austausch strukturierter Daten im Internet. Gleichzeitig zeichnet sich in der Automatisierungstechnik der Bedarf ab mit Hilfe des Internets Prozessdaten weltweit verfügbar zu machen. Im vorliegenden Beitrag werden die Gründe und Chancen hierfür aufgezeigt und ein Ansatz vorgestellt, wie sich dies mit Hilfe von XML bewerkstelligen lässt. Dabei werden zum einen Aufbau und Funktionsweise einer Website zur Überwachung technischer Prozesse beschrieben, in der die Prozessdaten in Form von XML-Dokumenten ausgetauscht werden. Zum anderen wird eine mögliche Struktur vorgeschlagen, in der Prozessdaten dargestellt werden können, und gezeigt, wie sich daraus XML-Dokumente mit Prozessdaten gewinnen lassen.

## 1 Einführung

Der Einsatz globaler Kommunikationsinfrastrukturen in der Automatisierungstechnik bietet zahlreiche technologische und wirtschaftliche Vorteile. Von lesenden und schreibenden Zugriffen auf einzelne Prozessdaten oder Parameter (z.B. Temperaturwert, Messbereichsgrenzen von Sensoren) bis hin zu komplexen Bedien- und Wartungsabläufen, all diese Interaktionen können nicht mehr nur an Ort und Stelle des Automatisierungssystems sondern von beliebigen entfernten, u.U. sogar weltweit verteilten Rechnern aus durchgeführt werden. Gleichzeitig lässt sich die Kluft zwischen Automatisierungs- und Geschäftswelt überbrücken: Produktions- und Lagerhaltungszahlen sind nicht mehr Vergangenheitswerte vom Vortag oder noch weiter zurück liegenden Zeitpunkten, sondern können sofort und unmittelbar, d.h. „online" ermittelt werden.

Angesichts dieser Perspektiven zeichnet sich derzeit der Trend ab das Internet als Medium für die benötigte weltweite Kommunikation von Prozessdaten nutzbar zu machen. Das Internet bietet eine besonders kostengünstige Möglichkeit zur globalen Kommunikation, da es keine Neuverlegung spezieller Kabelnetze erforderlich macht, sondern plattform- und netzwerkübergreifend die bestehenden Netzwerkinfrastrukturen zu einem globalen Informationskanal verknüpft. Der Zugang kann mit vergleichsweise billigen Standardkomponenten bewerkstelligt werden. Abgesehen davon sprechen noch einige weitere Gründe dafür Prozessdaten über das Internet zugänglich zu machen. Zum einen ist auf Grund der Homogenität der

Kommunikation über alle Bereiche hinweg zu erwarten, dass sich die Integration von Prozessdatenzugriffen und den darauf aufbauenden Applikationen in bestehende Systeme und Geschäftsabläufe erheblich vereinfacht. Ein weiterer wesentlicher Aspekt ist die Vereinheitlichung der Handhabung und Bedienung. Entwicklung, Inbetriebnahme und Benutzung von Anwendungen zur Prozessdatenverarbeitung können mit Hilfe bewährter und verbreiteter Konzepte und Werkzeuge (z.B. Webbrowser) erfolgen ohne teure Spezialisten oder lange Einarbeitungszeiten in Kauf nehmen zu müssen.

Voraussetzung für Anwendungen dieser Art ist, dass geeignete Konzepte und Technologien verfügbar sind, die einen Austausch von Prozessdaten über das Internet möglich machen. Zwar hat sich das Internet mit großem Erfolg als Verbreitungskanal für Dateien und für von Menschen lesbaren Dokumenten bewährt, weniger aber für von Maschinen lesbaren Daten. Speziell stehen mit dem Internet die erforderlichen Technologien für den Zugang (z.B. HTTP, FTP, usw.) und die Anzeige (z.B. HTML) von Informationen bereit. Nur in geringem Maße haben sich bisher jedoch Mechanismen etabliert, die auch eine über die reine Informationsanzeige hinaus gehende Verarbeitung von Daten, oder im vorliegenden Fall von Prozessdaten, erlauben. Gerade hier in diesem Punkt verspricht die extensible Markup Language (XML) [9] Abhilfe, die als plattformunabhängiger Standard für die Repräsentation und den Austausch von Daten im Internet immer mehr an Bedeutung gewinnt. Anders als in einigen Ansätzen, die auf Mechanismen zum Aufruf von Methoden entfernter Objekte über das Internet beruhen [3], erfolgt bei XML die Datenübertragung mit Hilfe von textbasierten Dokumenten, sehr ähnlich zur Art und Weise, wie auch zur Anzeige bestimmte Informationen in Form von HTML-Dokumenten transportiert werden. Bereits in vielen Bereichen wird XML eingesetzt, insbesondere die sog. Business to Business (B2B) Anwendungen, d.h. Anwendungen zum Austausch von Geschäftsdaten über das Internet, sind ein gutes Beispiel hierfür.

Im vorliegenden Beitrag wird gezeigt, wie XML für den Zugriff auf Prozessdaten angewendet werden kann. Dies wird am Beispiel einer Website erläutert, die eine Visualisierung und Konfiguration von technischen Prozessen über das Internet gestattet. Des Weiteren wird ein Strukturierungsschema für Prozessdaten eingeführt, das als Grundlage für die Darstellung der Prozessdaten in XML-Dokumenten dient. Dabei wird so vorgegangen, dass die Prozessdaten nach ihrer Art und Herkunft unterschieden werden, und gezeigt, dass diese Informationen auch in der XML-Darstellung der Prozessdaten unverändert wiederzufinden sind.

Im nächsten Abschnitt wird zunächst am Beispiel des Anwendungsfelds der Automobilelektronik heraus gearbeitet, welche Perspektiven und Chancen in der Zukunft mit der die Internetanbindung technischer Prozesse verbunden sind. Die Grundfunktion aller in diesem Umfeld entstehenden Anwendungen ist der lesende und schreibende Zugriff auf Prozessdaten. Wie dies auf XML-Basis verwirklicht werden kann, zeigt der dritte Abschnitt, in dem Architektur und Funktionsweise einer Website zur Prozessüberwachung dargelegt werden. Im vierten Abschnitt wird eine Möglichkeit zur hierarchischen Strukturierung von Prozessdaten vorgestellt und anhand eines Beispiels gezeigt, wie daraus die XML-Darstellung von Prozessdaten abgeleitet werden kann.

## 2 Potenzial der Internetnutzung in technischen Prozessen

In vielen Bereichen ist zu beobachten, dass elektronische Systeme, die zur Steuerung technischer Prozesse eingesetzt werden, immer zahlreicher und komplexer werden. Dies ist hauptsächlich auf die ständig wachsende Leistungsfähigkeit der Elektronik zurückzuführen sowie die Kostenvorteile, die ein hoher Automatisierungsgrad mit sich bringt.

Ein Beispiel für diese Tendenz stellen elektronische Systeme im Automobilsektor dar. Bereits heute machen Elektronikkomponenten bei verschiedenen Herstellern je nach Ausstattung zwischen 20% bis 26% der Kosten eines Fahrzeugs aus, für das Jahr 2005 wird dieser Wert auf über 30% prognostiziert. Hinzu kommt hier die Bestrebung elektronische Systeme nicht mehr nur für Aufgaben im Komfortbereich (z.B. elektrischer Fensterheber) oder zur Unterstützung der Fahrfunktionen (z.B. Fahrdynamikregelung) einzusetzen. Vielmehr zeichnet sich ab, dass in der Zukunft mechanische Komponenten durch verteilte elektronische Systeme, sog. Drive-by-Wire Systeme [1], ersetzt werden. Dies bedeutet, dass auch die Fahrfunktionen an sich mit elektronischen Mitteln realisiert werden, was die Hersteller solcher Systeme im Bezug auf die Sicherheit und Verlässlichkeit vor eine neue Dimension von Herausforderungen stellt.

In der Folge fallen oftmals zusätzliche, vorab nicht absehbare Entwicklungs- und Testzeiten an, es kommt zu erheblichen Verzögerungen bei der Produkteinführung und vielfach sind zur nachträglichen Fehlerbehebung in bereits ausgelieferten Fahrzeugen teure Rückrufaktionen notwendig. Eine der Gegenmaßnahmen besteht in der Realisierung zusätzlicher Funktionen zur Diagnose und Beeinflussung der elektronischen Systeme. Damit werden die Voraussetzungen für eine vereinfachte Fehlererkennung geschaffen, für ein erleichtertes Ermitteln von Fehlerursachen sowie für die Feinabstimmung mit der hydraulisch-mechanischen Systemumgebung. Ziel ist es Test und Inbetriebnahme zielgerichteter und effizienter abwickeln zu können und damit letztendlich die Komplexität der elektronischen Systeme beherrschbar zu machen sowie die Entwicklungskosten zu senken.

Vor dem Hintergrund der Tatsache, dass die Möglichkeit zur globalen Kommunikation über das Internet in immer mehr Bereichen an Bedeutung gewinnt, bietet die zunehmende Diagnose- und Beeinflussungsfunktionalität in der Automobilelektronik noch ein Potenzial in ganz anderer Hinsicht, nämlich die Durchführung von Diagnose- und Wartungsarbeiten an Fahrzeugen von entfernten Orten aus. Es eröffnet sich eine große Vielfalt an denkbaren Szenarien. Beispielsweise könnte eine Unterstützung im Pannenfall wie folgt aussehen: Ein Servicetechniker loggt sich von seiner Dienststelle aus in das liegen gebliebene Fahrzeug ein, ermittelt Art und Ursache des Fehlers und kann bereits binnen Minuten entscheiden, ob eine Reparatur vor Ort möglich ist, ob ein langsames Weiterfahren bis zur nächsten Werkstatt in Frage kommt oder ob abgeschleppt werden muss. Weiter könnten manche der alljährlichen Routineüberprüfungen von entfernten Servicebetrieben aus durchgeführt werden, während das betreffende Fahrzeug in der Garage zu Hause verbleibt, die Werkstatt bräuchte erst aufgesucht werden, nachdem ein konkreter Reparaturbedarf festgestellt wurde. Auf gleichem Wege ließen sich auch die Kosten bei der nachträglichen Fehlerbehebung in bereits ausgelieferten

Fahrzeugen drastisch senken, indem an Stelle der heute üblichen Rückrufaktionen Software-Updates per Fernzugriff ins Fahrzeug eingespielt werden.

## 3 Website zur Prozessüberwachung

Eine Teilaufgabe, die mit Blick auf die im vorangehenden Abschnitt angerissene Zielsetzung anfällt, besteht in der Visualisierung und der Konfiguration von technischen Systemen über das Internet. Am Institut für Automatisierungs- und Softwaretechnik der Universität Stuttgart wurde ein Konzept zur Entwicklung von Websites ausgearbeitet, mit denen diese Tätigkeiten vom Webbrowser aus durchgeführt werden können. Die Kernidee besteht darin, dass für den erforderlichen Transfer von Prozessdaten über das Internet XML-Dokumente benutzt werden. Im diesem Abschnitt werden Funktionsweise und Architektur einer solchen XML-basierten Website zur Prozessüberwachung vorgestellt. Anschließend wird erläutert, wie dieses Konzept für ein Beispiel aus dem Bereich der Automobilelektronik in die Praxis umgesetzt wurde.

### 3.1 Funktionsmerkmale

Mit der Website zur Prozessüberwachung können über das Internet verschiedene Ansichten für unterschiedliche Gruppen von Prozessdaten abgerufen werden. Dabei werden zwei Arten von Ansichten unterschieden: Visualisierungs- und Konfigurationsansichten.

Visualisierungsansichten dienen zur reinen Beobachtung von technischen Prozessen. Es wird eine bestimmte Auswahl von Prozessdaten zur Anzeige gebracht und Änderungen können „online" mitverfolgt werden. Den umgekehrten Fall stellen die Konfigurationsseiten dar. Hier kann der Benutzer durch Eingabe von Parametern das Prozessverhalten gezielt beeinflussen und verändern. Zusätzlich werden auch hier gewisse Prozessdaten angezeigt um die Wirkung vorgenommener Parameteränderungen überprüfen zu können.

In beiden Fällen müssen zwischen Prozess und Website-Ansicht Daten ausgetauscht werden. Dabei erfolgen sowohl lesende als auch schreibende Zugriffe auf Prozessdaten. Im Folgenden werden Prozessdaten, die vom Prozess in eine Ansicht heruntergeladen werden, als Diagnosedaten bezeichnet, umgekehrt heißen Prozessdaten, die von einer Ansicht in der Prozess hochgeladen werden, Konfigurationsdaten.

In Visualisierungsansichten werden fortlaufend die benötigten Diagnosedaten vom Prozess angefordert und die Anzeigefelder dementsprechend aktualisiert. Das Zeitintervall zwischen den Ansichtsaktualisierungen hängt von der im Netzwerk zur Verfügung stehenden Bandbreite ab. In Konfigurationsansichten erfolgt die Übermittlung von Konfigurationsdaten an den Prozess, sobald der Benutzer eine Eingabe von Prozessparametern getätigt hat.

### 3.2 Umgebung

Die Umgebung, in der die Website zur Prozessüberwachung betrieben wird, ist in Abbildung 1 dargestellt.

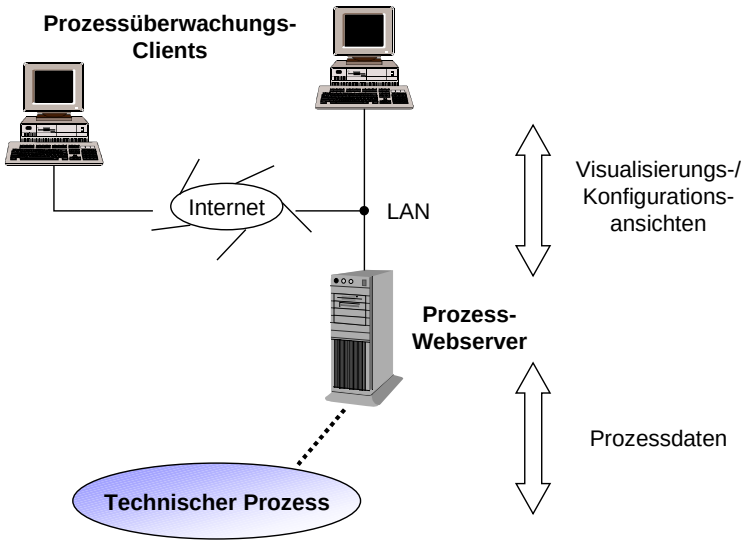


**Abb. 1.** Technischer Prozess, Prozess-Webserver und Prozessüberwachungs-Clients

Den zentralen Kern der Website zur Prozessüberwachung bildet der Prozess-Webserver. Er hält sämtliche Visualisierungs- und Konfigurationsansichten zum Abruf bereit und übernimmt die Abwicklung der Prozessdatenzugriffe. Ein solcher Prozess-Webserver muss eigens für den zu überwachenden Prozess bereitgestellt werden und kann somit als integrales Bestandteil von technischen Prozessen mit Internetanbindung betrachtet werden.

Der Datenaustausch zwischen Prozess-Webserver und Prozess kann sich auf unterschiedliche Weisen vollziehen. Zum einen kann hierfür ein gesonderter Prozesszugriffskanal vorgesehen werden [5]. Es kommen hierfür z.B. eine Ethernet- oder CAN- oder RS485-Verbindung in Frage. Darüber hinaus wird ein geeignetes Prozesszugriffsprotokoll benötigt, das Adressen und Nachrichtenformate für das Lesen und Schreiben der Prozessdaten definiert. Diese Form des Prozessdatenzugriffs ist immer dann sinnvoll, wenn der zu überwachende Prozess als eine geschlossene Einheit angesehen werden kann.

Liegt im Gegensatz hierzu der zu überwachende Prozess als verteiltes System vor, oder verfügt er über ein Feldbussystem, so kann ein separater Prozesszugriffskanal entfallen. Stattdessen kann der Prozess-Webserver als weiterer Teilnehmer im Kommunikationsnetzwerk des verteilten Systems bzw. als zusätzlicher Feldbusteilnehmer realisiert werden [4].

Das Gegenstück zum Prozess-Webserver bilden die Prozessüberwachungs-Clients. Hierfür können beliebige Rechner mit Internet-Zugang herangezogen werden. Sie

greifen auf den Prozess-Webserver zu, bringen die dort verfügbaren Prozessdatenansichten zur Anzeige und bilden den Kommunikationsendpunkt der Diagnose- und Konfigurationsdatenübermittlung. Anzahl und Standort der Prozessüberwachungs-Clients, die mit dem Prozess-Webserver in Verbindung stehen können, sind nicht festgelegt. Sie können sich sowohl im selben lokalen Netz wie der Prozess-Webserver sowie auch an einem beliebigen anderen Ort befinden, der über das Internet zugänglich ist. Ebenso besteht die Möglichkeit, dass mehrere Prozessdaten-Clients gleichzeitig Anfragen an den Prozess-Webserver richten.

### 3.3 Architektur

In der Website zur Prozessüberwachung wird XML eingesetzt um den Prozessdatenaustausch zwischen Prozess-Webserver und Prozessüberwachungs-Clients zu bewerkstelligen. Sowohl auf Client- als auch auf Serverseite müssen daher Softwarekomponenten eingesetzt werden, die eine hinreichende Unterstützung für die Verarbeitung von XML-Dokumenten bieten. Derzeit kann unter den Anbietern von Internet-Software die Fa. Microsoft in ihren Produkten die bisher weitreichendste XML-Unterstützung vorweisen. Diese Tatsache war maßgebend für die Festlegung der Architektur sowie die Auswahl der Softwarekomponenten in Prozess-Webserver und Prozessüberwachungs-Client (vgl. Abbildung 2).

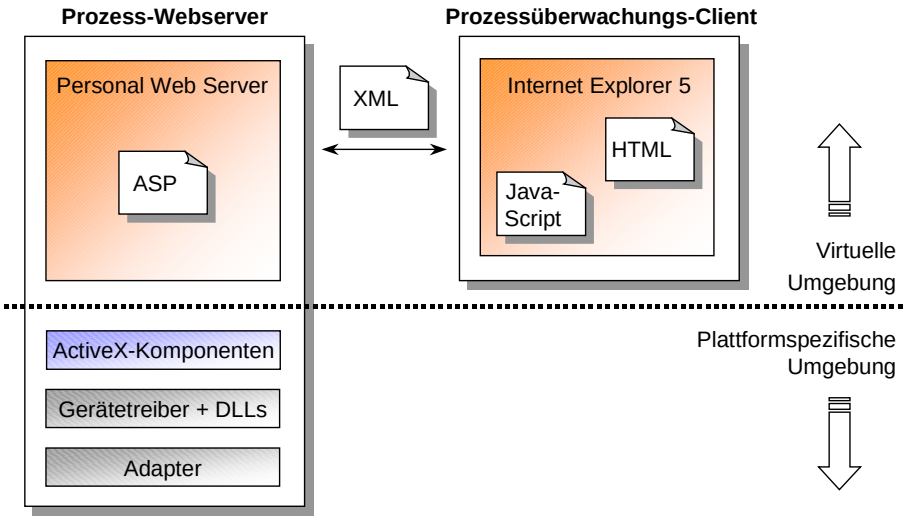


**Abb. 2.** XML-fähiger Prozess-Webserver und Prozessüberwachungs-Client

**XML-fähiger Prozess-Webserver.** Der Prozess-Webserver besteht aus einem handelsüblichen PC mit Windows NT 4.0 als Betriebssystem. Je nach Form des Prozessdatenzugriffs muss ein entsprechender Kommunikationsadapter einschließlich passendem Gerätetreiber installiert sein. Als Webserver-Software können der MS Internet Information Server 4.0 oder alternativ der MS Personal Web Server benutzt werden. Damit alle erforderlichen Softwarekomponenten für die XML-Verarbeitung verfügbar sind, muss darüber hinaus der MS Internet Explorer 5 installiert sein.

Auf einem solchermaßen konfigurierten Prozess-Webserver können im Wesentlichen zwei Arten von Dokumenten zum Abruf bereitgestellt werden: HTML-Seiten und sog. Active Server Pages (ASP) [8]. Die HTML-Seiten können neben formatiertem Text, Grafik und anderen multimedialen Inhalten auch ausführbaren Programm-Code in Form von Skripten enthalten. Dieser wird nach dem Abrufen der betreffenden HTML-Seite auf Clientseite ausgeführt. ASP-Seiten beinhalten in erster Linie ebenfalls Programm-Code in Skriptform, die Ausführung erfolgt jedoch nicht auf Client- sondern auf Serverseite. Zur Erstellung von Skriptcode kommen sowohl server- als auch clientseitig eine ganze Reihe von Skriptsprachen in Frage, z.B. Java Script [2], Visual Basic Script, u.a..

Der Prozess-Webserver soll Prozessdaten in Form von XML-Dokumenten im Internet bereitstellen bzw. von dort entgegennehmen. Hierbei gilt es den Umstand zu berücksichtigen, dass zur Prozessseite hin sämtliche ein- und ausgehenden Prozessdaten über den dafür vorgesehenen Kommunikationsadapter laufen. Letzterer ist Hardware-Bestandteil des Prozess-Webservers. Sämtliche Zugriffe seitens der Software, so auch das Lesen und Schreiben der Prozessdaten, müssen über den im System installierten Gerätetreiber erfolgen. Folglich sind für den Zugang zu den Prozessdaten auf dem Prozess-Webserver immer plattformspezifische Zugriffe auf Systemebene notwendig.

Demgegenüber laufen alle Aktivitäten, die für Interaktionen mit entfernten Rechnern im Internet relevant sind, in einer sog. virtuellen Umgebung ab. D.h. der Abruf von Dokumenten sowie auch die Ausführung von Skripten erfolgen unabhängig von zu Grunde liegender Hardware und installiertem Betriebssystem. Somit können für Client- und serverseitige Skripte ausschließlich plattformunabhängige Skriptsprachen eingesetzt werden, die von Haus aus keine Zugriffe auf System- bzw. Hardwareebene unterstützen.

Eine der Möglichkeiten, um eine Verbindung zwischen der virtuellen Welt auf Webserver-Ebene und der plattformspezifischen Welt auf System- bzw. Hardwareebene herzustellen, bietet der Einsatz von sog. ActiveX-Komponenten. Hierbei handelt es sich um eine von der Fa. Microsoft entwickelte und für Windows-Plattformen spezifische Technologie, die auf dem ebenfalls von dort stammenden Konzept des Common Object Model (COM) [6] beruht. Jede ActiveX-Komponente enthält eine oder mehrere Klassen, die jeweils eine bestimmte Funktionalität verkörpern. Die Schnittstellen dieser Klassen werden durch Eigenschaften, Methoden und Ereignisse gebildet.

Entscheidend für die Eignung zum Durchgriff vom World Wide Web auf PC-Hardware und -System sind folgende zwei Merkmale des ActiveX-Konzepts. Zum einen können in ActiveX-Klassen plattformspezifische Operationen ausgeführt werden. Damit sind in ActiveX-Klassen jederzeit auch System- und Hardwarezugriffe möglich. Zum anderen lassen sich ActiveX-Komponenten in beliebigen Programmier-

und Skriptsprachen verwenden, d.h. auch in Skriptsprachen, die in Webapplikationen eingesetzt werden. Als Einschränkung darf jedoch nicht unerwähnt bleiben, dass ActiveX-Komponenten und Applikationen, die sie verwenden, ausschließlich auf Windows-Plattformen ausführbar sind.

**XML-fähiger Prozessüberwachungs-Client.** Prinzipiell kann jeder Rechner als Prozessüberwachungs-Client eingesetzt werden, solange er über einen Internetzugang verfügt und außerdem ein XML-fähiger Webbrowser installiert ist. Da in der Praxis bisher jedoch nur der MS Internet Explorer 5 als Webbrowser mit integrierter XML-Unterstützung verfügbar ist, ist man derzeit noch auf dieses Produkt und die davon unterstützten Plattformen eingeschränkt.

Die Hauptaufgabe des Prozessüberwachungs-Client besteht im Anzeigen der abgerufenen HTML-Dokumente. Sofern vorhanden, wird darin enthaltener Skriptcode ausgeführt, außerdem werden Benutzereingaben abgewickelt.

### 3.4 Realisierung

Mit der vorstehend beschriebenen Architektur sind die notwendigen technologischen Voraussetzungen gegeben um eine Website zur Prozessüberwachung zu realisieren und dabei den Prozessdatenaustausch mit XML abzuwickeln. Im weiteren muss überlegt werden, aus welchen Bestandteilen die einzelnen Prozessdatenansichten aufgebaut werden können und wie zu verfahren ist um die Aktualisierung der dargestellten Prozessdaten zu bewerkstelligen.

**Bestandteile von Prozessdatenansichten.** In jeder Visualisierungs- und Konfigurationsansicht der Website für Prozessüberwachung kommen drei Klassen von Bestandteilen vor: Dokumente, Skripte und Objekte.

Als Dokumente werden zum einen HTML-Seiten benötigt. Jene bilden das Front-End der Website zur Prozessüberwachung und beinhalten alle statischen Daten einer Prozessdatenansicht. Dies sind im Wesentlichen Überschriften und Beschriftungstexte für Prozessdatenelemente sowie sonstige gestalterische Elemente um der Ansicht ein ansprechendes Aussehen zu verleihen. Des Weiteren sind an entsprechenden Stellen Platzhalter vorgesehen um die variablen Ansichtsdaten, d.h. die Prozessdaten selbst darzustellen.

Einen weiteren Dokumentbestandteil von Prozessdatenansichten stellen die XML-Dokumente dar, die zum Transport der Prozessdaten zwischen Client und Server eingesetzt werden. Sie treten auf sowohl in Form von XML-Anfragedokumenten, die ein Prozessüberwachungs-Client zum Prozess-Webserver übermittelt werden, als auch in umgekehrter Richtung als XML-Antwortdokumente.

Um das Senden und Empfangen von XML-Dokumenten sowie deren Verarbeitung zu ermöglichen, ist Skriptcode erforderlich. Da solche Verarbeitungsschritte sowohl auf Client- sowie auch auf Serverseite stattfinden müssen, tragen zu jeder Prozessdatenansicht sowohl clientseitige Skriptanteile innerhalb von HTML-Dokumenten als auch serverseitig ausgeführte ASP-Seiten bei.

Die Objekte der Prozessdatenansichten kommen auf dem Prozess-Webserver zum Einsatz. Es handelt sich dabei um Instanzen von Klassen der dort installierten ActiveX-Komponenten. Ihr Zweck besteht in der Übermittlung von Prozessdaten aus bzw. in den Prozess. Folglich finden innerhalb dieser Objekte sämtliche Funktionsaufrufe statt, die zur Bedienung des für den Prozessdatenzugriff verwendeten Kommunikationsadapters erforderlich sind.

**Aktualisierung von Prozessdatenansichten.** Die Aktualisierung der Prozessdaten verläuft in Visualisierungs- und Konfigurationsansichten jeweils unterschiedlich. Visualisierungsansichten erfordern ein zyklisch wiederholtes Herunterladen von Diagnosedaten. Demgegenüber muss in Konfigurationsansichten nach erfolgter Benutzereingabe ein Hochladen von Konfigurationsdaten stattfinden.

Beide Abläufe, d.h. Diagnosedaten-Download und Konfigurationsdaten-Upload, werden auf Initiative des Prozessüberwachungs-Clients hin angestoßen (vgl. Abbildung 3). Der in den HTML-Seiten eingebettete Skriptcode tritt in Aktion, es wird ein XML-Anfragedokument generiert und an den Prozess-Webserver gesendet. Beim Diagnosedaten-Download ist darin der Name der betreffenden Prozessdatenansicht enthalten, beim Konfigurationsdaten-Upload dagegen die vom Benutzer getätigten Eingaben.

Auf dem Prozess-Webserver übernimmt Skriptcode, der in Form von ASP-Seiten vorliegt, die Auswertung des empfangenen XML-Anfragedokuments. Im Falle des Diagnosedaten-Downloads werden in Abhängigkeit des erhaltenen Ansichtsnamens die benötigten Diagnosedaten aus dem Prozess ermittelt. Umgekehrt erfolgt beim Konfigurationsdaten-Upload die Übergabe der übermittelten Konfigurationsdaten an den Prozess. Hierzu werden jeweils entsprechende ActiveX-Objekte für den Prozessdatenzugriff erzeugt und angewendet. Schließlich wird ein XML-Antwortdokument generiert und an den Prozessüberwachungs-Client zurückgesendet. Handelt es sich um einen Diagnosedaten-Download, so finden sich hierin die gewonnenen Diagnosedaten. Ein Konfigurationsdaten-Upload wird dagegen mit einem leeren XML-Dokument beantwortet.

Auf dem Prozessüberwachungs-Client wartet der in den HTML-Seiten eingebettete Skriptcode auf den Empfang des XML-Antwortdokuments. Daraufhin wird als letzter Schritt eines Diagnosedaten-Downloads die Aktualisierung der betreffenden Prozessdatenansicht vorgenommen. Dies geschieht durch Auslesen der Diagnosedaten im XML-Antwortdokument und Eintragen in die zugeordneten Platzhalter der HTML-Seiten. Demgegenüber sind bei einem Konfigurationsdaten-Upload an dieser Stelle keine weiteren Verarbeitungsschritte mehr erforderlich.

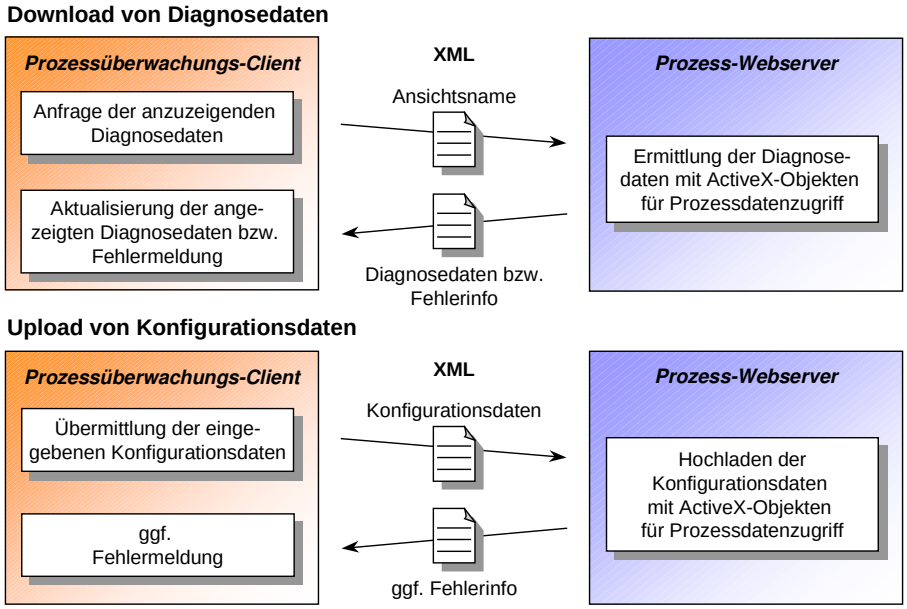


**Fig. 3.** Diagnosedaten-Download und Konfigurationsdaten-Upload

## 4 Struktur für XML-Dokumente mit Prozessdaten

In den vorangehenden Abschnitten wurden Aufbau und Funktionsweise einer Website zur Prozessüberwachung dargestellt, in der XML-Dokumente zum Transport von Prozessdaten zwischen Client und Server eingesetzt werden. Im Folgenden wird der Inhalt von solchen XML-Dokumenten betrachtet. Dazu wird eine Möglichkeit aufgezeigt, wie die Diagnosedaten in XML-Antwortdokumenten bzw. die Konfigurationsdaten in XML-Anfragedokumenten mit Hilfe der Sprachmittel von XML darstellbar sind.

Ähnlich wie das bereits seit längerem verbreitete HTML bietet XML die Möglichkeit in Dokumenten neben den eigentlichen Daten zusätzlich sog. Tags zur Markierung der Daten abzulegen [7]. Der zur Verfügung stehende Vorrat an Tags ist jedoch nicht fest vorgegeben und beschränkt, sondern es können beliebige eigene Tags definiert und benutzt werden.

Eine der Stärken von XML beruht darauf, dass sich Daten nicht als nur als Einzelwerte sondern gesammelt in Gruppen darstellen lassen und dazu hierarchisch strukturiert werden können. Dies geschieht mit Hilfe der Tags, deren Inhalt aus einem Datenwert oder aus weiteren untergliederten Tags bestehen kann. Damit lassen sich beliebige Gruppierungen und Schachtelungen von Datenelementen erzielen. Gleichzeitig geht aus dem Namen der Tags eine Information über die Art des jeweils enthaltenen Datenelements oder der Datengruppe hervor.

Um Prozessdaten in XML darzustellen, muss somit zunächst überlegt werden, auf welche Weise Prozessdaten gruppiert und hierarchisch untergliedert werden können. Im vorliegenden Fall erfolgt dieser Schritt unter Annahme der Zielsetzung, dass aus der hierarchischen Struktur der Prozessdaten gewisse Informationen über deren Eigenschaften ersichtlich sein sollen.

### 4.1 Hierarchieschema für Prozessdaten

Den Ausgangspunkt für die Hierarchisierung der Prozessdaten bildet der einfachste Fall einer Prozessdatengruppe: einzelne Prozessdatenwerte (z.B. Temperaturwert, Regelparameter, usw.) werden zu Prozessdatensätzen zusammengefasst. Als Kriterium für die Einteilung in Prozessdatensätze dient die logische Zusammengehörigkeit der Prozessdaten (z.B. Motordrehzahl und Motordrehmoment). Es wird jedoch vereinbart, dass alle Prozessdatenwerte eines Prozessdatensatzes stets von derselben Quelle im Prozess stammen.

An den so entstehenden Prozessdatensätzen lassen sich folgende Eigenschaften feststellen. Zum einen können Prozessdatensätze je nach Anzahl und Art der enthaltenen Prozessdatenfelder verschiedene Typen aufweisen. Weiter wird angenommen, dass der Prozess als verteiltes System vorliegt und Prozessdaten sowie auch Prozessdatensätze i.A. von verschiedenen Quellen im Prozess stammen.

Prozessdaten können damit als Mengen von Prozessdatensätzen aufgefasst werden, in denen Prozessdatensätze verschiedener Typen und mit verschiedenen Quellen im Prozess vorkommen. Auf Basis dieser Überlegungen kann folgendes Hierarchieschema für Prozessdaten festgelegt werden:

- Prozessdatenfeld: einzelner Prozessdatenwert. Bilden zusammen die unterste Hierarchiestufe.
- Prozessdatensatz: logisch zusammengehörende Gruppe von Prozessdatenfeldern, die alle von derselben Quelle im Prozess stammen. Anzahl und Art der enthaltenen Prozessdatenfelder legen den Typ des Prozessdatensatzes fest.
- Prozessdatensatzliste: Menge von Prozessdatensätzen gleichen Typs und mit derselben Quelle im Prozess. Dies ist z.B. der Fall, wenn eine bestimmte Gruppe von Messwerten chronologisch erfasst und aufgezeichnet wird.
- Prozessdatensatzsammlung: Menge von Prozessdatensätzen gleichen Typs und mit verschiedenen Quellen im Prozess. Eine denkbare Situation für diesen Fall wäre z.B. ein fehlertolerant ausgelegtes System, das mit zwei redundanten Rechnerknoten ausgestattet ist und somit an zwei verschiedenen Orten die gleichen Prozessdatensätze verarbeitet.
- Prozessdatenmenge: Menge von Prozessdatensätzen verschiedenen Typs und mit verschiedenen Quellen im Prozess. Dies ist die oberste Hierarchiestufe und stellt die allgemeinste Sichtweise auf Prozessdaten dar. Ihr können jegliche Prozessdaten untergliedert werden, die in einem verteilten Prozess vorkommen.

Aus einem solchen Hierarchieschema für Prozessdaten lässt sich auf einfache Weise deren XML-Darstellung gewinnen. Dies soll im Folgenden anhand eines Beispiels verdeutlicht werden.

### 4.2 Abbildung der Prozessdaten auf XML

Zur praktischen Erprobung des hier vorgestellten Konzepts wurde am Institut für Automatisierungs- und Softwaretechnik der Universität Stuttgart eine Website zur Überwachung des Modellprozesses IAS-Kart realisiert (vgl. Abbildung 4). Es handelt sich um einen Gokart, an dem die mechanische Lenkstange entfernt und durch ein voll-elektronisches Lenksystem ersetzt wurde. Diese exemplarische Steer-By-Wire Anwendung besteht aus den verteilten Funktionseinheiten Lenkwunscherfassung, Radwinkelmessung und Servomotoransteuerung. Zusammen bilden sie einen Regelkreis, der über einen Servomotor fortlaufend den Einschlag der Räder entsprechend dem aktuellen Lenkwunsch einstellt (vgl. Abbildung 5).

**Abb. 4.** Modellprozess IAS-Kart

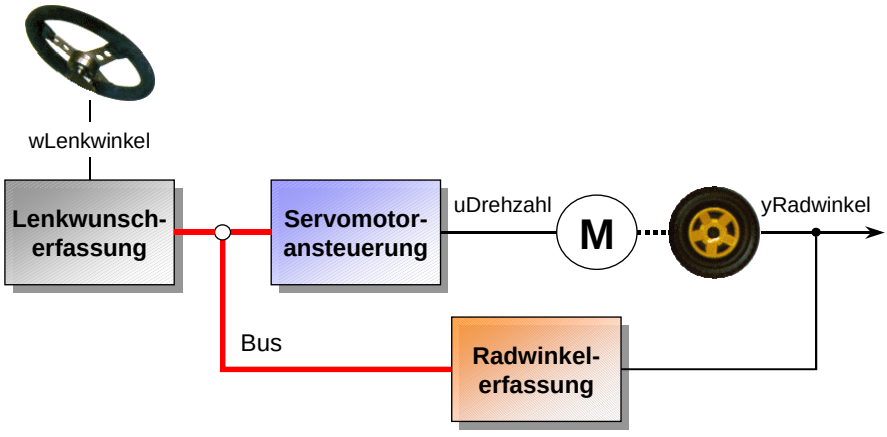


**Abb. 5.** Systemmodell der Steer-By-Wire Anwendung

In der zugehörigen Website zur Prozessüberwachung ist u.a. eine Visualisierungsansicht Echtzeitdaten abrufbar. Neben anderen Werten sind hierin die Prozessdaten der Radwinkelmessung und der Servomotoransteuerung ersichtlich. Beide Prozessdatengruppen setzen sich jeweils aus verschiedenen Einzelwerten zusammen. Im speziellen Fall der Radwinkelmessung sind dies der Radwinkel und die Radwinkeländerung. Im fehlertolerant ausgelegten elektronischen Lenksystem wurden für jede der drei Funktionseinheiten zwei redundante Rechnerknoten vorgesehen. Folglich sind auch die Radwinkel- und Servomotordaten in jeweils zwei unterschiedlichen Quellen vertreten.

Bei Anwendung des Hierarchieschemas aus dem vorangehenden Abschnitt würde ein XML-Antwortdokument für das Herunterladen der hier beschriebenen Prozessdaten wie in Abbildung 6 dargestellt aussehen:

```
<?xml version="1.0"?>
<RealTimeData>
  <ClusterWheelData>
    <NodeWheelData NodeName="FSU1">
      <WheelDataItem>
        <Angle>30</Angle>
        <AngleChange>5</AngleChange>
      </WheelDataItem>
    </NodeWheelData>
    <NodeWheelData NodeName="FSU2">
      ...
    </NodeWheelData>
  </ClusterWheelData>
  <ClusterServoMotorData>
    ...
  </ClusterServoMotorData>
</RealTimeData>
```

**Abb. 6.** XML-Antwortdokument mit Echtzeitdaten des IAS-Kart

Wie sich unschwer erkennen lässt, deckt sich die Struktur der Tags im XML-Dokument exakt mit o.g. Hierarchieschema. Der Wurzel-Tag RealTimeData repräsentiert die Prozessdatenmenge, der alle betrachteten Echtzeitdaten des IAS-Karts untergliedert sind. Die darin eingeschlossenen Tags ClusterWheelData und ClusterServoMotorData sind Prozessdatensatzsammlungen für den Typ von Prozessdatensätzen, den die Funktionseinheiten Radwinkelmessung bzw. Servomotoransteuerung jeweils anliefern. Wegen der zwei redundanten Rechnerknoten pro Funktionseinheit untergliedern sich diese in jeweils zwei Prozessdatensatzlisten. Im Falle der Radwinkeldaten sind dies die Tags NodeWheelData einschließlich nachfolgendem Attribut zur Kennzeichnung des Quellrechnerknotens. Die dort vorhandenen Ausgaben der Radwinkeldaten sind die mit den Tags WheelDataItem eingegrenzten Prozessdatensätze. Innerhalb davon befinden sich schließlich zwei Prozessdatenfelder als Bestandteile der

Radwinkeldaten. Sie setzen sich zusammen aus den Tags Angle und AngleChange sowie dem jeweiligen Prozessdatenwert als Elementinhalt.

Ergänzend zeigt Abbildung 7 das Erscheinungsbild der Visualisierungsansicht Echtzeitdaten, in der die Informationen aus dem auszugsweise vorgestellten XML-Dokument betrachtet werden können.

**Abb. 7.** Visualisierungsansicht Echtzeitdaten

Das Vorgehen zur Repräsentation von Prozessdaten in XML lässt sich damit wie folgt zusammenfassen: Zunächst muss ein Hierarchieschema gefunden werden, in dem die Prozessdaten zusammenhängend dargestellt werden können. Die Entwicklung eines solchen Schemas erfolgt unter Zuhilfenahme der Eigenschaften - hier Art und Herkunft - der Prozessdaten. Die so gewonnene Hierarchie bildet zusammen mit applikationsspezifisch gewählten Benennungen die abstrakte Semantik der Prozessdaten und lässt sich 1:1 auf XML abbilden. In der Folge entstehen in jedem XML-Dokument mit Prozessdaten einheitliche Tags und Abfolgen bzw. Schachtelungen derselben. Überdies sind mit dem Hierarchieschema automatisch auch die Bedeutungen bekannt, die in XML-Dokumenten mit Prozessdaten durch Tags zum Ausdruck gebracht werden.

## 5 Schlussfolgerungen

Die Nutzung von Internettechnologien in der Automatisierungstechnik eröffnet ein breites Feld neuartiger Anwendungsmöglichkeiten und wird sich in der nächsten Zeit in allen Bereichen mehr oder weniger durchsetzen. Bereits heute werden hierzu in

Forschung und Industrie viele Aktivitäten in Gang gesetzt, manche sprechen sogar vom Anfang eines revolutionsartigen Umbruchs.

Schlüsselrolle innerhalb dieser Entwicklung wird diejenige Technologie einnehmen, die sich als Standard zur Repräsentation von Daten im Internet etablieren wird. XML stellt in dieser Richtung einen viel versprechenden Ansatz dar. In diesem Beitrag wurde gezeigt, wie sich diese Technologie auch für den weltweiten Zugriff auf Prozessdaten einsetzen lässt. Im Besonderen wurde am Beispiel einer Website zur Visualisierung und Konfiguration technischer Prozesse erläutert, auf welche Weise gelesene und zu schreibende Prozessdaten in Form von XML-Dokumenten transportiert werden können.

Darüber hinaus wurde deutlich gemacht, dass in XML-Dokumenten Daten, und damit auch Prozessdaten, gesammelt in und strukturierter Form vorliegen. Alle notwendigen Informationen über Datenart und -struktur lassen sich durch Tags zum Ausdruck bringen, die zur Kennzeichnung der Datenelemente und -gruppierungen verwendet werden. Da man hierbei auf keinen feststehenden Tag-Satz beschränkt ist, sondern beliebige eigene Tags erlaubt sind, ist auch das Strukturierungsschema zur Darstellung von Informationen in XML keineswegs fest vorgegeben, sondern kann passend zum Problem gewählt werden. Dies wurde im vorliegenden Beitrag dazu ausgenutzt um ein beispielhaftes Hierarchieschema für Prozessdaten so zu definieren, dass u.a. eine Unterscheidung der Prozessdaten nach Art und Herkunft möglich ist. Als Folge werden in XML-Dokumenten mit Prozessdaten durch die Tags genau diese Information wiedergespiegelt.

Neben den im Rahmen dieses Beitrags diskutierten Punkten gibt es noch eine Reihe von anderen triftigen Gründen, die für den Einsatz von XML zum Austausch von Prozessdaten über das Internet sprechen. Einer davon ist die Unabhängigkeit von der Applikation. Ausschlaggebend hierfür ist das Prinzip der strikten Trennung der Struktur und des Inhalts eines Dokuments von seinem Lay-out. Damit lassen sich die in XML-Dokumenten transportierten Prozessdaten gleichzeitig und an verschiedenen Orten in völlig unterschiedlichen Anwendungen nutzen. Einer der möglichen Anwendungsfälle ist die Darstellung von Prozessdaten im Webbrowser, als andere denkbare Verwendungszwecke kämen z.B. der hardware in the loop-Betrieb von Projektierungs- und Debugwerkzeugen, die automatische Auswertung von Produktionsdaten, die Speicherung in Datenbanken, usw. in Frage.

Ein weiterer bemerkenswerter Aspekt von XML ist die Plattformunabhängigkeit. Zwar bietet derzeit die Fa. Microsoft den einzigen Webbrowser mit integrierter XML-Unterstützung an, es ist jedoch zu erwarten, dass in Bälde vergleichbare Produkte anderer Softwarehersteller am Markt erscheinen werden. Grundsätzlich ist die Verarbeitung von XML-Dokumenten auf jedem Rechner- und Betriebssystem möglich. Auch das zur Übertragung verwendete Protokoll, bisher zumeist HTTP oder FTP, bleibt letzten Endes frei wählbar. Auf Grund der Tatsache, dass die Datenübertragung mit XML nicht binär sondern in Textform erfolgt, bleiben Prozessdaten, die als XML-Dokument vorliegen, für bestehende und zukünftige Anwendungen zugänglich und das ohne Betriebssystem, Übertragungsprotokoll oder Programmiersprache in irgendeiner Form vorschreiben zu müssen.

Zu den Aspekten, die in der Zukunft behandelt werden sollen, zählen u.a. der Einsatz von XML für eine über das Lesen und Schreiben hinausgehende Verarbeitung von Prozessdaten (z.B. Alarm- & Ereignisbehandlung, chronologisches Aufzeichnen

von Prozessdaten). Auch wird ein besonderes Augenmerk darauf gerichtet sein, wie sich ausgehend von bestimmten Ausgangsinformationen, wie z.B. der physikalischen Struktur eines Automatisierungssystems, das passende Hierarchieschema für Prozessdaten und damit auch deren XML-Darstellung gewinnen lassen. Darauf aufbauend soll untersucht werden, wie eine Integration von unterschiedlich strukturierten Prozessdaten sowie von Prozessdaten und Informationen aus anderen Bereichen vorgenommen werden kann.

## Literatur


1. Brite-EURAM III: X-By-Wire, Safty Related fault tolerant Systems in Vehicles, EU-Forschungsprojekt, 1996 - 1998.
2. Holzner, S.: Java Script Complete, McGraw-Hill, USA, 1998.
3. Jazdi, N.: Ansteuerung eines industriellen Kaffeeautomaten per Internet, GMA-Fachtagung „Industrielle Automation und Internet/Intranet-Technologien", Langen, 1999.
4. Kopetz, H.; Kucera, M.; Millinger, D.; Ebner, C.; Smaili, I.: Interfacing Time-Triggered Embedded Systems to the INTERNET, International Symposium on Internet Technology, Taipei, Taiwan, 1998.
5. Kucera, M.; Smaili, I.; Fuchs, E.: A Lightweight Ethernet Protocol to Connect a Time-Triggered Real-Rime System to an INTERNET Server, European Multimedia, Microprocessor Systems and Electronic Commerce Conference and Exhibition, Bordeaux, Frankreich, 1998.
6. Rogerson, D.: Inside Microsoft's Component Object Model, Microsoft Press, USA, 1997.
7. Spencer, P.: XML Design and Implementation, Wrox Press, Groß-Britannien, 1999.
8. Williams, A.; Barber, K.; Newkirk, P.: ACTIVE SERVER PAGES BLACK BOOK: The Professional's Guide to Developing Dynamic, Interactive Web Sites with Microsoft ActiveX, Coriolis Group Books, USA, 1998.
9. World Wide Web Consortium: Extensible Markup Language (XML) 1.0, Technical Report REC-xml-19980210, W3C, 1998.