

Multi-User-Webinterface unter Verwendung von LabVIEW

(Fachartikel VIP 2003)

Andreas Himmelreich, Prof. Dr.-Ing. Jörg Müller
 Fachhochschule Jena, Fachbereich Elektrotechnik/Informationstechnik

Kurzfassung

Die Grenzen zwischen der Automatisierungstechnik und Informationstechnik verwischen zunehmend. Der Grund dafür liegt in der Verwendung der Technologien der IT-Welt in der Automation. Derzeit wird vielerorts speziell der Einsatz der Webtechnologie vorangetrieben. Webtechnologie heißt für den Automatisierer, auf eine Palette von bewährten, allgemein akzeptierten und weit verbreiteten Techniken zurück greifen zu können. Ziel des Einsatzes dieser Technologie soll es sein, den industriellen Kommunikationsstrukturen mehr Transparenz zu verleihen und derzeit vorhandene Kommunikationsbarrieren verschwinden zu lassen.

Diese Tendenzen innerhalb der Branche waren der Anlass dafür, ein Webinterface unter LabVIEW™ zu realisieren. Die konkrete Aufgabe bestand darin, Front Panels auf entfernten Rechnern mittels Standardwebbrowser zu betrachten und zu bedienen. Dem Anwender soll es so ermöglicht werden, seine LabVIEW-Applikation via Internet/Intranet fernzusteuern.

Durch den Einsatz objektorientierter Mechanismen, wie z. B. Multiinstanzfähigkeit und Skalierbarkeit, sowie der Ereignissteuerung entstand ein Multiuser-System bei dem gleichzeitig der Verbrauch der Rechnerressourcen und der Netzwerktraffic minimiert wurde. Berücksichtigt wurde auch, dass das Publizieren von VI's ohne aufwendige Erweiterungen des Programmcodes ermöglicht wird.

Abstract

The limits between automation engineering and information engineering blur increasingly. The reason for it lies in the use of the technologies of the IT world in the automation. At present in many places especially the input of the World Wide Web technology is driven forward. World Wide Web technology means for the automation technician to be able to access on a pallet of techniques which are approved, accepted generally and widespread. Aim of the input of this technology shall be to give more transparency to the industrial communication structures and at presently existing communication barriers to dispose of. These trends within the line of business were the cause to realize a World Wide Web interface under LabVIEW™. The concrete task consisted in looking and using front panels on remote computers by means of standard Web browsers. It shall be made possible for the user to operate its LabVIEW™ application by remote control via internet/intranet. By the use of object-oriented mechanisms, as the multi instance ability and event control, resulted a multinational user system, which minimized the consumption of computer resources and the net work traffic at the same time. It was also taken into account that publishing is made possible by VI's without consuming expansions of the program code.

Anwendungsfall

Mit einem einfachen Beispiel soll zunächst gezeigt werden, wie einfach man seine Applikation webfähig gestalten kann. Als Anwendungsbeispiel wurde ein Fall aus der Gebäudeautomation gewählt, speziell die Steuerung eines Raumes.

Die Steuerungselemente des Raumes (Taster, Lampen, Fensterkontakte, Temperatursensoren, ...) sind auf I/O-Klemmen geführt, die wiederum über einen Feldbus mit einem Feldbuscontroller verbunden sind, auf dem ein Steuerungsprogramm läuft. Über einen OPC-Server stehen sämtliche Items einer übergeordneten Steuerung zur Verfügung. Als übergeordnete Steuerung soll ein einfaches VI erstellt werden.

Im Ergebnis ist es möglich per Standardwebbrowser auf einem entfernten Rechner über das Internet/Intranet diesen Raum entsprechend seinen Möglichkeiten zu steuern.

Es ergibt sich folgende Struktur:

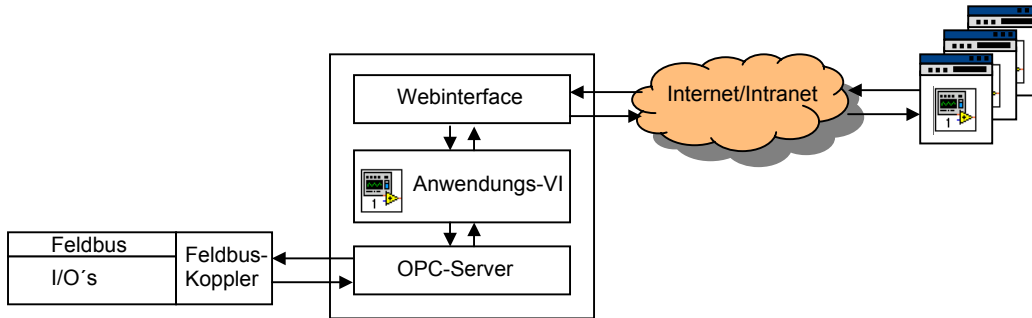


Bild 1: Struktur der Beispielanwendung

Im ersten Schritt muss das Anwendungs-VI erstellt werden, welches die Funktionen des Raumes abbildet. Dazu wird zunächst die Bedienoberfläche mit den benötigten Controls und Indikatoren entworfen. Diese werden über eine DataSocket Connection mit dem OPC-Server verbunden. Die Option „Browse for Item“ ermöglicht dabei die schnelle und komfortable Zuordnung der Elemente. Im Anschluss muss im Programmcode lediglich dafür gesorgt werden, dass bei Änderung einer der Werte ein Event für das Webinterface ausgelöst wird, um in diesem Fall den Client zu aktualisieren. Eine Aktualisierung nach jedem Schleifendurchlauf, was einem Update mit dem OPC-Server entspricht, würde zu unnötigem Verbrauch der Rechnerressourcen und erhöhtem Netzwerktraffic führen.

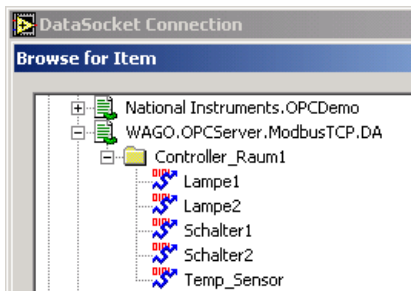


Bild 2: Browsen der Items des OPC-Servers

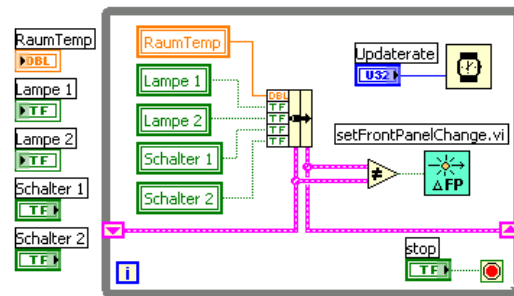


Bild 3: Einfügen der Eventquelle

In letzten Schritt muss das fertige VI für den entfernten Zugriff noch publiziert werden. Dazu wird einfach der Publisher des Webinterfaces aufgerufen. Dieser erzeugt die erforderliche HTML-Seite mit den VI-spezifischen Parametern. Mit dem Aufruf dieser Seite kann nun die Steuerung des Raumes von einem entfernten Rechner aus erfolgen.

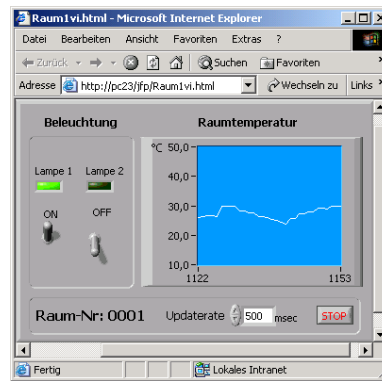
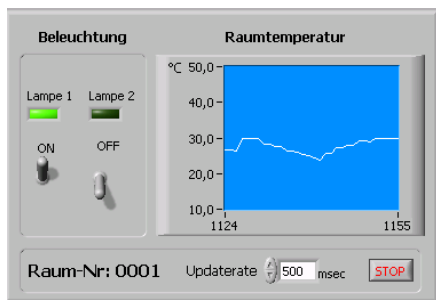


Bild 4: Anwendungs-VI und der Zugriff mittels Webbrowser im Intranet

Funktionsprinzip

Durch Verwendung der Webtechnologien ergibt sich zwangsläufig eine Client-Server-Struktur für die Anwendung. Die Grundbausteine sind der Webbrowser als Client und ein Webserver. Damit ist man zunächst in der Lage, sich die in den Verzeichnissen des Webserver gespeichert HTML-Seiten anzeigen zu lassen. Seit der Version 6i ist standardmäßig ein Webserver in LabVIEW integriert, so dass die Grundvoraussetzung bereits vorhanden sind. Für eine Interaktion mit LabVIEW-VI's sind jedoch client- und serverseitig zusätzliche Funktionen erforderlich.

Die Client-Seite

Mit einem Webbrowser als Container für die Client-Anwendung stößt man fast zwangsläufig auf Java. Java ist plattformunabhängig und die gängigen Webbrowser unterstützen Java als Plug-in. Eine weitreichende Verbreitung ist damit sichergestellt. Weiterhin findet man zahlreiche Funktionen (Java-Klassen) zur schnellen und unkomplizierten Netzwerkprogrammierung, sowie umfangreiche Möglichkeiten der Interaktion. Nicht nur deswegen wird Java auch als die Sprache des Internets bezeichnet. Clientseitig wird also eine Java-Anwendung in Form eines Java-Applets eingesetzt. Das Java-Applet erfüllt folgende Aufgaben:

- Verbinden mit dem Webinterface über die konfigurierten Ports,
- Darstellung des Frontpanels,
- Entgegennehmen und Weiterleiten von Bedienhandlungen.

Die Server-Seite

Die Server-Anwendung bildet das eigentliche Webinterface und hat folgende Aufgaben zu realisieren:

- Publizieren von VI's für den entfernten Zugriff,
- Eingehende Verbindungswünsche feststellen,
- Zugriffskontrolle,
- Handeln der Verbindungen zwischen Client und VI,
- Einstellung von Anwendungsparametern.

Aufbau der Verbindung

Für jedes veröffentlichte VI existiert eine HTML-Datei. Durch Eingabe der URL im Browser wird diese über den Webserver in den Browser geladen. Der Applettag signalisiert dem Browser, dass weiterhin ein Applet zu laden ist. Das Applet übernimmt die Parameter aus der HTML-Seite und baut die Verbindung zum Webinterface auf. Es kann nun das jeweils aktuelle Front Panel empfangen und Bedienhandlungen gesendet werden.

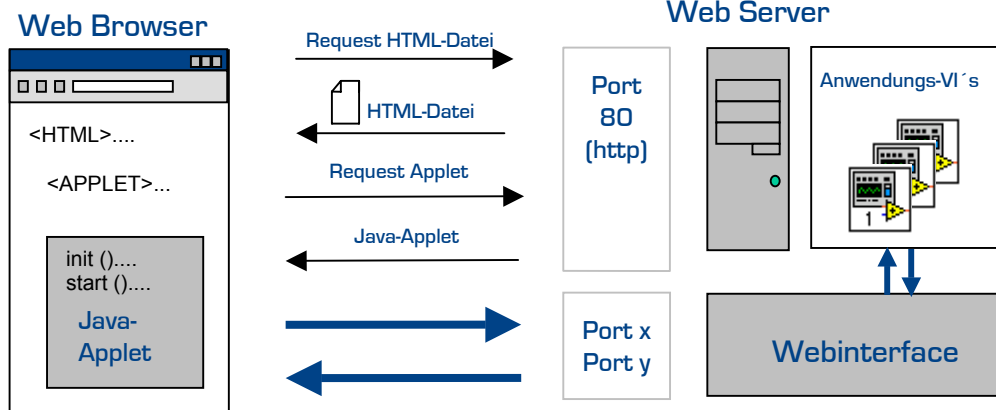


Bild 5: Funktionsprinzip

Das Webinterface

Die Umsetzung des Webinterface erfolgt mit LabVIEW 6.1 in Kombination mit dem Object Event Toolset (OET). Das Object Event Toolset ist Bestandteil von ObjectVIEW™, einer Entwicklung der Firma Vogel Automatisierungstechnik GmbH. Dieses Toolset ermöglicht die Umsetzung objektorientierter Mechanismen in LabVIEW.

Die Realisierung erfolgte in drei Ebenen.

- Graphisches User Interface (GUI),
- Server-Funktionen,
- Connection Handler.

Die jeweiligen Funktionseinheiten wurden als unabhängige Objekte abgebildet, welche über Messages und Notifications ereignisgesteuert miteinander kommunizieren. In einer CASE-Struktur sind die Methoden des Objektes definiert. Diese können über Ereignisse von anderen Objekten (public) oder nur vom Objekt selbst (privat) aufgerufen werden. Sind keine Ereignisse vorhanden, verweilt das Objekt im Ruhezustand und verbraucht keine Rechnerleistung.

GUI-Objekt

Das GUI-Objekt ermöglicht die Interaktion mit dem Nutzer des Webinterfaces. Es startet (Online-Modus) und stoppt (Offline-Modus) das Serverobjekt. Im Online-Modus werden alle aktuellen Verbindungen mit Statusinformationen, den verbundenen Clients und angeforderten VI's in einer Liste angezeigt. Dazu wird bei sich ändernden Verbindungsverhältnissen im Serverobjekt die Methode „InfoUpdate“ im GUI-Objekt aufgerufen und die Liste aktualisiert.

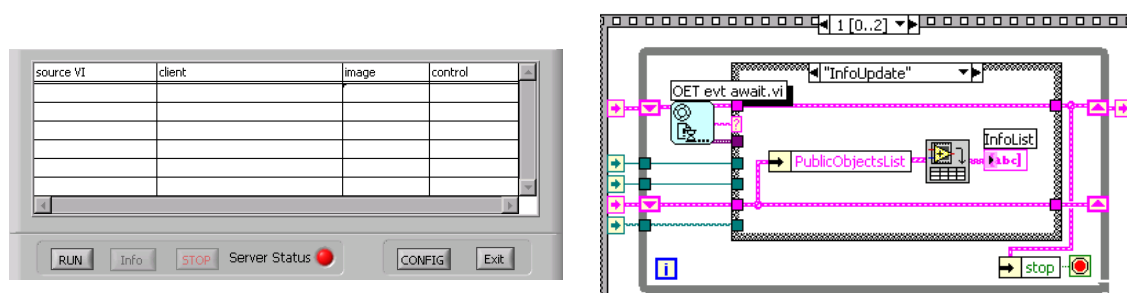


Bild 6: Frontpanel des Webinterfaces und die Methode „InfoUpdate“

Im Offline-Modus kann das Webinterface konfiguriert werden. Dazu gehört:

- Einstellung der Webinterface-Parameter,
- Publizieren von VI's,
- Verwaltung der Zugriffs- und Bedienrechte.

Sämtliche Konfigurationsdaten werden in einer gemeinsamen Datei im XML-Format abgespeichert.

Serverobjekt

Im Serverobjekt werden die jeweils konfigurierten Ports auf eingehende Verbindungen abgehört. Wird ein Verbindungswunsch festgestellt, wird die Methode „NewConnection“ im Server-Objekt aufgerufen. Diese überprüft zunächst die Verbindung auf Gültigkeit. Dazu wird die im GUI-Objekt angelegte Zugriffsliste geparkt. In Abhängigkeit des Überprüfungsergebnisses wird die neue Verbindung wieder geschlossen oder ein neuer ConnectionHandler instanziiert und gestartet. Jede eingehende Verbindung erhält somit ein unabhängiges Handler-Objekt zugewiesen.

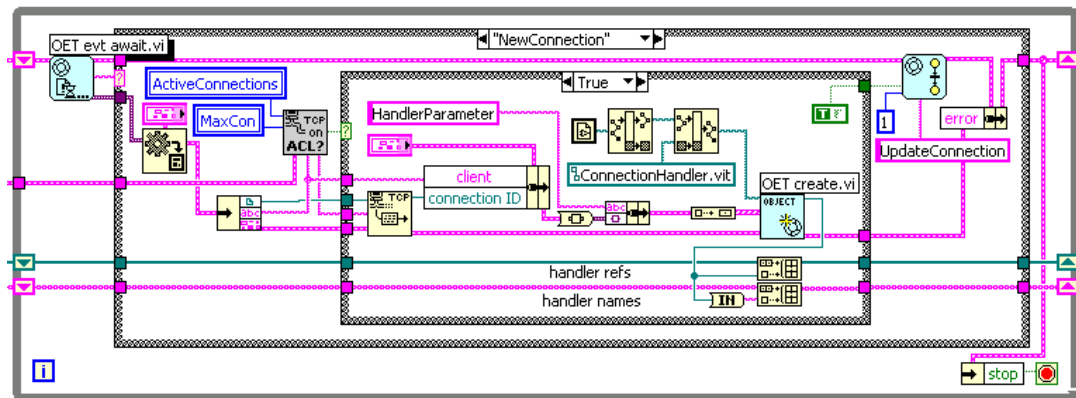


Bild 7: Die Methode „NewConnection“ im Server-Objekt

Handlerobjekte

Die Handlerobjekte bilden die unterste Ebene in der Hierarchie. Jeder aktiven Verbindung ist ein Handlerobjekt zugeordnet. Die Handlerobjekte reagieren auf Ereignisse (MouseEvent, KeyEvent) des Applets und leiten diese über die Windows-API an die entsprechenden VI's. Umgedreht wird bei Änderungen des Frontpanels das gerade aktuelle Frontpanel an das Applet übertragen. Wird das Applet gestoppt, z.B. weil das Browserfenster geschlossen wurde, wird die Verbindung beendet und es erfolgt der Aufruf der Methode „DelConnection“ im Serverobjekt. Dort wird der Handler ausgetragen und danach die Methode „InfoUpdate“ im GUI-Objekt aufgerufen, um die Liste der aktuellen Verbindungen zu aktualisieren. Das Handlerobjekt selbst wird zerstört und gibt seine Ressourcen wieder frei.

Zusammenfassung und Ausblick

Im Ergebnis steht nun eine Anwendung, die es ermöglicht ohne großen Installations- und Programmieraufwand Applikationen webfähig zu gestalten. Die Vorteile der grafischen Programmierung unter LabVIEW erstrecken sich damit auch auf das Erstellen von Webapplikationen. Dem Anwender eröffnet sich auf einfachste Weise das weite Feld des WWW, eine vertraute und intuitiv zu bedienende Technologie. Automatisierungsaufgaben können schnell und mit minimalem Kostenaufwand auf andere Plattformen portiert und quasi weltweit verteilt werden. Sämtliche Clients, die eine Unterstützung von Java bieten, sind einsetzbar, wie z.B. Pocket-PC´s. Geräte werden eine eigene Homepage erhalten, über die die Bedienung oder im Fehlerfall eine Ferndiagnose möglich ist. Neue Anwendungsfelder tun sich auf und es erweitert sich das Dienstleistungsspektrum des LabVIEW-Programmierers.

Literatur

[ObjV02] Vogel Automatisierungstechnik GmbH, ObjectVIEW- Handbuch,
URL: <http://www.objectview.de>