

Konzeption und Umsetzung einer Browserapplikation zur Steuerung und ereignisbezogenen Visualisierung industrieller Prozesse

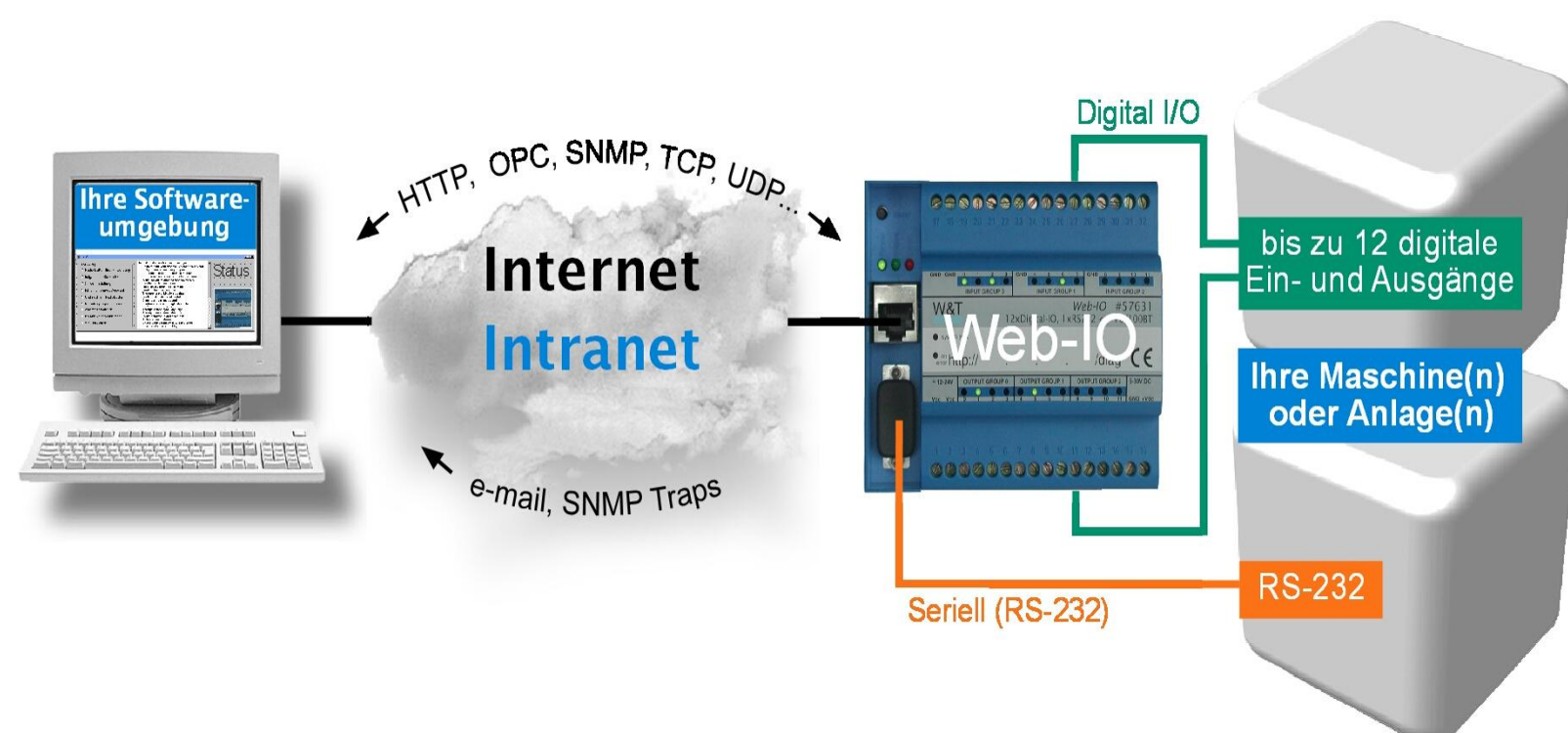
Bachelorarbeit, vorgelegt von David Walter

Aufgabenstellung

Das Ziel dieser Arbeit ist es, Konzepte zu erarbeiten, welche die Umsetzung einer asynchronen Kommunikation zwischen einer Browserapplikation und dem vorhandenen Webio-Server ermöglichen.

Die verschiedenen Varianten sind zu untersuchen, um eine Entscheidung für die praktische Umsetzung zu treffen. Anschließend gilt es die Anforderungen für eine Software zu ermitteln, welche die Steuerung und die ereignisbezogene Visualisierung eines industriellen Prozesses ermöglicht. Nach diesen Anforderungen soll ein Prototyp entwickelt werden, welcher zur Erprobung des praktischen Einsatzes dient.

Web-IO



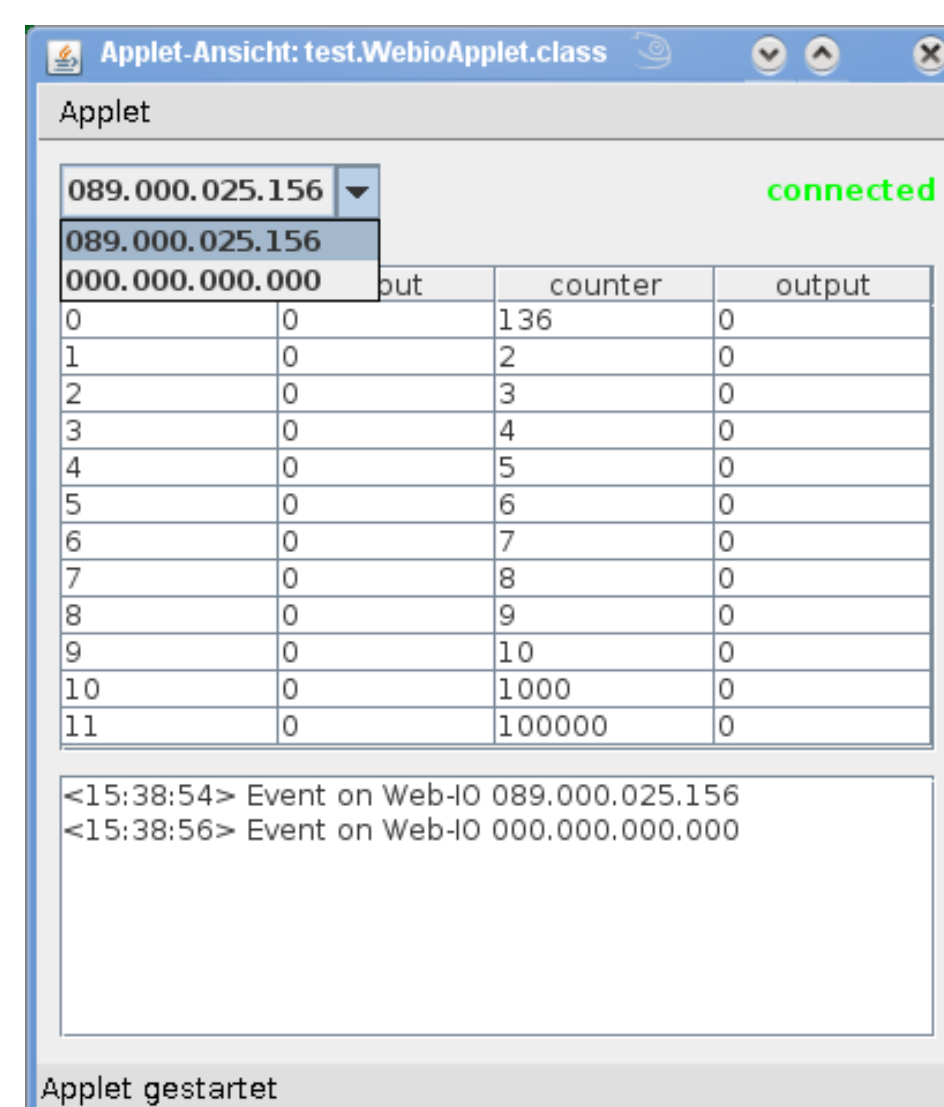
Web-IO 12xDigital Input, 12xDigital Output, RS-232 57631
(Quelle: Wiesemann & Theis GmbH www.wut.de)

Ein Web-IO stellt die Schnittstelle zwischen digitalen Ein- und Ausgängen, seriellen Schnittstellen und dem Ethernet dar. Bei dem verwendeten Gerät handelt es sich um das Web-IO 12x Digital Input 12x Digital Output mit RS-232-Com-Server-Funktionalität. Ein solches Gerät besitzt ein Webinterface, welches dem Auslesen und der Steuerung der Ein- und Ausgänge, sowie auch der Konfiguration dient. Außerdem unterstützt es Netzwerkprotokolle für die Anbindung individueller Software. Mit einem Web-IO ist es möglich industrielle Maschinen an ein Netzwerk anzubinden und mit einer individuellen Software zu verwalten und zu steuern.

Webio-Server

Der Webio-Server ist ein Multithreaded-Server und dient zur Verwaltung von Web-IO-Geräten. Der Server nimmt die Signale der an das Netzwerk angeschlossenen Geräte entgegen und wertet diese aus. Dabei werden Änderungen von Eingängen und Zählern mit Hilfe einer Datenbank protokolliert und je nach Ereignis werden unter anderem Ausgänge geschaltet, Zählerstände ausgelesen oder Zähler zurückgesetzt. Außerdem ist es möglich eine manuelle Beschaltung der Ausgänge vorzunehmen. Die Kommunikation zwischen dem Server und den Endgeräten wird über TCP-Verbindungen realisiert. Über die Konfiguration des Servers kann die Funktionsweise der Web-IO-Geräte an individuelle Prozesse angepasst werden.

Webio-Applet

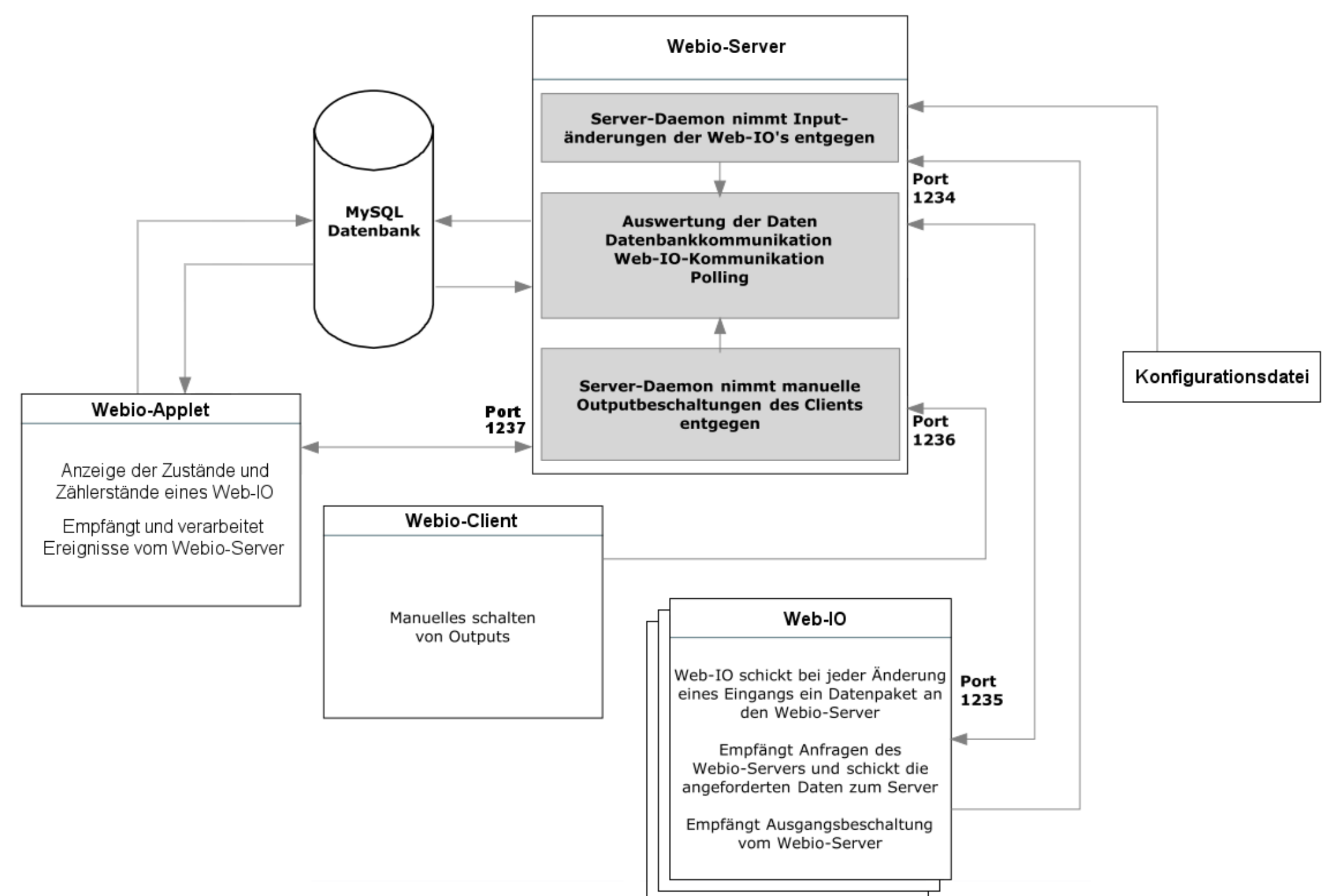


Die Entscheidung für die Umsetzung der Browserapplikation fiel nach Betrachtung von mehreren Varianten auf ein Java-Applet. Ein Java-Applet hat geringe Anforderungen an die Client- und Server-Seite, funktioniert plattformunabhängig, lässt sich in eine HTML-Seite einbinden und ermöglicht eine ereignisbezogene Visualisierung der Web-IO-Daten.

Das Applet kommuniziert über eine TCP-Verbindung mit dem Webio-Server und über eine JDBC-Schnittstelle mit der Datenbank. Damit ein Applet die Rechte für die Initialisierung der beiden benötigten Verbindungen erhält, wird es mit Hilfe der Tools keytool und jarsigner signiert. Das Zertifikat muss vom Anwender akzeptiert werden, damit das Applet ausgeführt werden kann.

Über die Combobox können die verschiedenen angeschlossenen Web-IO-Geräte ausgewählt werden, um die aktuellen Zählerstände, die Eingänge und die Beschaltung der Ausgänge einzusehen. Diese Daten werden in der Tabelle dargestellt. Das Event-Log unter der Tabelle zeigt eingegangene Ereignisse an.

Funktionsweise



Das Webio-Applet verbindet sich bei mit dem Webio-Server und der Datenbank. Dabei werden zunächst die IP-Adressen der Web-IO-Geräte aus der Datenbank gelesen. Außerdem wird eine Verbindung zum Webio-Server hergestellt und auf eingehende Ereignisse gewartet. Beim Statusänderungen an Web-IO-Geräten und der damit verbunden Verarbeitung durch den Server, wird das Applet durch den Server informiert. Daraufhin kann eine Aktualisierung der angezeigten Daten durchgeführt werden.

Fazit

Die Arbeit hat mehrere Varianten ergeben, eine ereignisgesteuerte Visualisierung als Browserapplikation umzusetzen. So wären auch eine Ajax-Anwendung, eine Java-Anwendung über Java Web Start und Umsetzungen mit Silverlight oder Flash denkbar gewesen. Das durch dieser Arbeit entstandene Java-Applet ist ein funktionierender Prototyp, der nun im Praxiseinsatz erprobt werden kann