

Web-HMIs versus native HMIs

Die Anwendung entscheidet

HMI-Software auf Basis reiner Web-Technik liegt im Trend, doch es gibt nach wie vor viele Anwendungen, in denen native HMI-Ansätze ihren Web-gestützten Pendanten überlegen sind.

Welche sind das, und wann sind Kombinationen sinnvoll?

Stefan Niermann, verantwortlich für den Vertrieb beim HMI-Software-Hersteller Inosoft, gibt Auskunft.

Markt&Technik: Welche Vor- und Nachteile haben Web-HMI-Konzepte, welche native HMI-Konzepte generell?

Stefan Niermann: Das native Konzept hat den großen Vorteil, dass man auf der Zielplattform wirklich alles nutzen kann, was sie bietet. Für unsere HMI-Software „VisiWin“ sage ich immer: »Damit kann man im HMI alles machen, was ein Windows-Rechner leistet.« Das liegt in dem Fall am offenen Ansatz unseres Systems, lässt sich aber auch auf andere Hersteller übertragen, soweit deren Produkte einen ähnlichen Funktionsumfang bieten.

Ein Vorteil des Browsers ist, dass in fast jedem Gerät mit Netzwerkanschluss und Display ein solcher installiert ist. Unterstützt der Browser die benötigten Standards, kann er einen Client für ein HMI ohne weitere Installation ausführen.

Ein wichtiger Unterschied ist, dass ein Browser eher den Charakter einer Anzeige und Bedienung hat. Im Gegensatz dazu kann der native Client zusätzlich auch komplexeste Logiken ausführen. Der Browser kann zwar JavaScript interpretieren, aber die Logik gehört eher zum Kontext der Seite und unterliegt aus verständ-



Typische Einsatzbereiche von HMI-Software – Design und Funktion richten sich nach Anwendung und Ziel-HMI-Gerät.

Bilder: Inosoft

lichen Sicherheitsgründen großen Einschränkungen beim Zugriff etwa auf Dateisystem, Betriebssystem und Hardware. Eine richtige Geschäftslogik gehört in diesem Szenario in den Server und wird dort vermutlich mit einer anderen Programmiersprache erstellt.

In welchen Anwendungen sind Web-HMI-Konzepte, in welchen native HMI-Konzepte und in welchen eine Kombination aus beiden am sinnvollsten?

Eine schöne Frage! Sie impliziert nämlich, dass man die Vor- und Nachteile jeder Lösung abwägen und dann das sinnvollste Konzept auswählen sollte. Ich höre immer wieder: »Das macht man doch heute so, das ist doch die Zukunft!« Die Entscheidung ist aus meiner Sicht sehr wichtig und sollte auf Basis von fundierten Fakten und nicht auf Basis von »das macht man doch so« getroffen werden.

Ein fest installiertes, lokales HMI an einer Maschine oder auch ein verteiltes System mit vielen Funktionen, einer ausgefeilten Oberfläche und hoher Funktionssicherheit ist häufig gut und geradlinig mit einer nativen Lösung umzusetzen.

Sobald ich ohne Installation auskommen muss, also beispielsweise Websites für den Betrieb, die Wartung oder die Reparatur einer Maschine mit einem mitgebrachten Tablet aufrufen will, bin ich mit der Browser-Lösung gut beraten. Übersichten für den Zustand der Maschine, Produktionszahlen oder ähnlich gelagerte Information kann ich ebenfalls im Browser gut darstellen. Eine Bedienung ist natürlich auch möglich. Da muss ich nur bei tragbaren Geräten auf die Sicherheit achten – man darf schließlich nichts bedienen, was man nicht sehen kann. Wenn der Kopf des Kollegen noch in der Maschine steckt ...

In vielen Fällen ist also die Kombination aus lokaler Bedienung mit einer nativen Lösung und zusätzlichen Geräten zur Übersicht, Wartung oder Überwachung mit dem Web-HMI eine sinnvolle Lösung. Die Botschaft lautet: Zu-

erst analysieren, welche Lösung am besten zu den Anforderungen passt!

Welche Möglichkeiten bieten Web-HMI- und native HMI-Konzepte in puncto Design und Grafik?

Aus technischer Sicht sind die grafischen Möglichkeiten beider Konzepte fast gleichwertig. Ich kann nur für uns sprechen, und da ist es so, dass wir beide Konzepte mit der jeweils besten geeigneten Technologie und absoluter Offenheit unterstützen. Natürlich hat

jedes System seine eigenen Stärken und Schwächen. Das würde aber nicht bedeuten, dass etwas gar nicht geht. Ich spreche hier allerdings nicht von extremen Anforderungen wie etwa in modernen High-End-Gaming-Anwendungen. Da ist der Browser im Nachteil.

Web-HMIs lassen sich mit Mobile Devices aller Art nutzen. Inwieweit muss das Design der Web-HMI-Oberfläche an unterschiedliche Mobile Devices angepasst werden?

Früher hatte man für diese Geräteklassen immer die speziell erstellten Websites, deren Adresse wie m.irgendwas.de lautete. Das macht man heute kaum noch, weil die Spannweite der Auflösungen und Formfaktoren mit einem festen Design nicht mehr sinnvoll abgedeckt werden kann. Also nutzen moderne Websites ein responsives Design. Wenn Sie zum Beispiel die Website www.inosoft.com im Vollbild öffnen und dann das Browser-Fenster immer schmaler machen, sehen Sie, dass die ganze Website sich verändert. Zuerst verschwindet das Hauptmenü, daraus wird ein sogenanntes Hamburger-Menü, also das mit den drei Strichen. Dann verschwinden große Grafiken, und so geht es weiter. Diese Schritte muss man individuell festlegen, und man braucht dafür spezielle Layout-Container, die das unterstützen.

Hier ist ein Wort der Warnung angebracht: Ich werde immer wieder gefragt, ob man nicht alles mit einer Oberfläche erledigen kann. Das ist nicht zu empfehlen. Eine komplexe Maschinenübersicht, ein R/I-Schema kann man nicht mal ansatzweise responsiv vom großen Bildschirm auf ein Smartphone herunterskalieren, sodass sie übersichtlich und gut bedienbar bleiben.

Webbrowser bekommen häufig Updates. Welche Folgen hat dies für Anbieter und Nutzer von Web-HMIs?

Das ist ein Punkt, der alle HMI/SCADA-Rechner in Firmennetzen und natürlich genauso Mobilgeräte betrifft. Eine hohe Aktualisierungsfrequenz ist durchaus ein Qualitätsmerkmal für einen Browser, weil dadurch bekannte Sicherheitslücken schnellstmöglich geschlossen werden. Dabei werden aber auch neue Funktionen und strategische Änderungen eingeführt. Damit meine ich beispielsweise, dass



Stefan Niermann, Inosoft

„ In vielen Fällen ist die Kombination aus lokaler Bedienung mit einer nativen Lösung und zusätzlichen Geräten zur Übersicht, Wartung oder Überwachung mit dem Web-HMI eine sinnvolle Lösung. “

Browser mutmaßlich bald kaum noch unverschlüsselte http-Verbindungen akzeptieren. Eines Morgens kommt dann der Werker an die Maschine, der Browser hat die neueste Version geholt und führt das HMI erst nach einer Meldungs- und Bestätigungsgorgie und dem Einrichten von Ausnahmeregeln wieder aus. Was sich da abspielt, können wir uns alle leicht vorstellen.

Um diesen Entwicklungen aus dem Weg zu gehen, kann ich vielleicht einen speziellen Browser verwenden, womit ich aber den Vorteil der installationslosen Verteilung des HMI verliere. Diesen Browser darf ich selbstverständlich nie zum Surfen im Web benutzen! Alternativ muss ich den Server des HMI immer wieder an die aktuellen Erfordernisse anpassen, damit alles weiter funktioniert. Das ist bei nativen Lösungen nicht erforderlich.

Inwieweit schränken Cyber-Security-Konzepte wie etwa Sandboxes die Funktionalität und Praktikabilität von Web-HMIs ein? Wie lässt sich dem Problem begegnen?

Das Sandbox-Konzept dient dazu, Angriffe von Schad-Software zu verhindern. Alles im Browser wird bei der Ausführung in der Sandbox so abgeschottet, dass aus ihr heraus kein Zugriff auf das Betriebssystem, auf Dateien oder auf die Hardware erfolgen kann. Solange ich mich im HMI auf reine Maschinenfunktionen beschränke, also beispielsweise auf Daten aus der SPS, ist das kein Problem. Will ich aber ein Gerät über einen USB-Port anbinden oder lokale Dateien bearbeiten, wird das von der Sandbox abgeschottet. Ich muss diese Aufgaben auf den Server-PC auslagern. Eine universelle Lösung gibt es nicht, man muss den Einzelfall betrachten.

Inwieweit ermöglichen Web-HMIs „Bring your own Device“-Konzepte, und welche Folgen ergeben sich daraus für die Cyber-Security?

Eigene private Geräte werden ziemlich sicher einen funktionsfähigen Browser mitbringen, weil die Nutzer den natürlich verwenden wollen. Von daher sollte der Einsatz gut möglich sein. Die Sicherheit ist dabei eine Schwachstelle. Im privaten Bereich wird auch gerne mal eine ältere Betriebssystemversion eingesetzt, weil es für das alte Handy keine Updates mehr gibt und das Gerät ja noch funktioniert. Neue Sicherheitslücken werden aber nur durch einen aktuellen Software-Stand gestopft, und damit ist das vertraute Gerät potenziell unsicher. Vielleicht sind auch Apps installiert, die im Datenbestand wildern oder das System angreifbar machen. Für IT-Administratoren eine Quelle für schlaflose Nächte.

Wie harmonisieren und kooperieren Web-HMI- und native HMI-Konzepte mit neuen Verfahren wie Cloud- und Edge-Computing? Inwieweit lassen sich Updates der HMI-Software über die Cloud organisieren?

Da sehe ich zunächst die Trennung zwischen der Datengewinnung und -aufbereitung sowie der reinen Darstellung. Ein Web-HMI ist auf keinen Fall zuständig für die Aufbereitung der Daten und deren Übermittlung in die Cloud. Das ist immer eine Server-seitige Aufgabe. Ein nativ umgesetztes HMI kann diesen Job parallel zur Bedienoberfläche mit übernehmen.

Wie schon erwähnt, sollte man die Anforderungen an das Vor-Ort-System analysieren und dann das bestgeeignete Konzept umsetzen. In dieses wird man dann den Teil integrieren, der



Als allgemeines Schutzkonzept für Browser verhindert die Sandbox beim Browser-basierten HMI zwar Angriffe von Schad-Software, bringt aber auch Einschränkungen mit sich, indem sie gewollte Zugriffe auf Dateien oder auf die Hardware unterbindet.

Daten direkt in die Cloud sendet oder im Sinne von Edge-Computing aufbereitet und dann versendet. Für die Anzeige der Ergebnisse aus der Cloud auf beliebigen Endgeräten ist eine Web-Lösung wahrscheinlich am besten geeignet.

Natürlich kann man die Ergebnisse aus der Cloud auch im lokalen HMI verwenden. Das wäre eine Kombination der Daten aus verschiedenen Quellen, lokal und Cloud. Diese Möglichkeit gibt es bei beiden Konzepten.

Ein Update aus der Cloud ist für ein Web-HMI am einfachsten, wenn man die erforderlichen Websites per Webserver zur Verfügung stellt. Dann müssen die Clients aber immer mit dem Cloud-Server verbunden sein. Alternativ kann man einen Update-Mechanismus für die HMIs in der Cloud bereitstellen. Das käme dann für beide HMI-Typen auf das Gleiche heraus.

*Das Interview führte
Andreas Knoll.*