

# Kein ‚Entweder-oder‘

Für Maschinenbauer stellt sich oft die Frage, wie sie Bedienschnittstellen realisieren sollen: Als komplett eigene Programmierung oder über eine der am Markt verfügbaren HMI-/SCADA-Lösungen zum reinen Parametrieren – beide Wege haben ihre Vor- und Nachteile. Und: Es gibt bereits Lösungen, die beide Welten verknüpfen.

Die Schnittstelle zwischen Mensch und Maschine hat einen entscheidenden Einfluss bei der Weiterentwicklung der industriellen Automation, in der immer mehr Funktionen und Optionen in Software realisiert werden. In erster Linie erlaubt ein HMI-/SCADA-System das Bedienen von Maschinen und Anlagen und gibt dabei Hilfestellung durch das Visualisieren verschiedener Zustände, Abläufe, Trends in der Maschine oder im Prozess. Größere Flexibilität der Anwendungen geht in der Regel mit höherer Komplexität von Maschinen und Anlagen einher. In Zeiten von Industrie 4.0 sind zudem umfangreichere Datenbestände aus digitalisierten Komponenten und zusätzlichen Sensoren verfügbar. In der Summe dieser Trends wird die Steuerung laufender Prozesse auf der einen Seite aufwendiger, auf der anderen Seite ergeben sich mehr Eingriffsmöglichkeiten, beispielsweise zum Optimieren der Produktion oder der präziseren Einhaltung

von Produktionsspezifikationen im Rahmen des Qualitätsmanagements.

Dementsprechend werden auch die Funktionen von HMI- und SCADA-Systemen immer umfangreicher. Assistenzsysteme für die Bedienung im laufenden Betrieb gehören ebenso zum Umfang wie die Unterstützung von Wartungs- und Reparaturaufgaben. Nicht zuletzt wird der fest installierte Bildschirm – an der Maschine oder im Leitstand – durch immer mehr Schnittstellen ergänzt. Zugriff aus der Ferne über das Internet sowie mobil über Notebook, Tablet oder Smartphone gehören für viele Anwender zum Standard. Darüber hinaus eröffnet das Aufkommen von Datenbrillen den Einsatz von Augmented und Virtual Reality (AR/VR), beispielsweise in der Wartung. Neue berührungslose Bedienkonzepte per Sprache, Gestik und Mimik sind in der Erforschung oder zum Teil schon in der Erprobung – ein Ende dieser Entwicklung ist nicht absehbar.

Bei all dem gilt es jedoch nicht nur die Einsatzmöglichkeiten für den Anlagenbetreiber und den Bediener an der Maschine auszuweiten. Die Anwendungsoberfläche muss trotz steigender Komplexität der Funktionen benutzerfreundlich bleiben. Denn Mängel bei der Usability können zu Fehlbedienungen führen, die unnötige Kosten verursachen oder gar die Sicherheit der Anlage oder der Bediener gefährden.

Dies alles ‚unter einen Hut‘ zu bringen, stellt eine echte Herausforderung für Maschinenbauer dar, der sie üblicherweise mit einer der folgenden Strategien begegnen: Zum einen kann ein Unternehmen die Aufgabe komplett in Eigenarbeit lösen und eine HMI- oder SCADA-Anwendung von Grund auf selbst programmieren. Zum anderen gibt es vorgefertigte Lösungen, die den Anforderungen über eine umfangreiche Parametrierung gerecht werden. Beide Wege haben ihre spezifischen Vor- und Nachteile, die man kennen sollte.

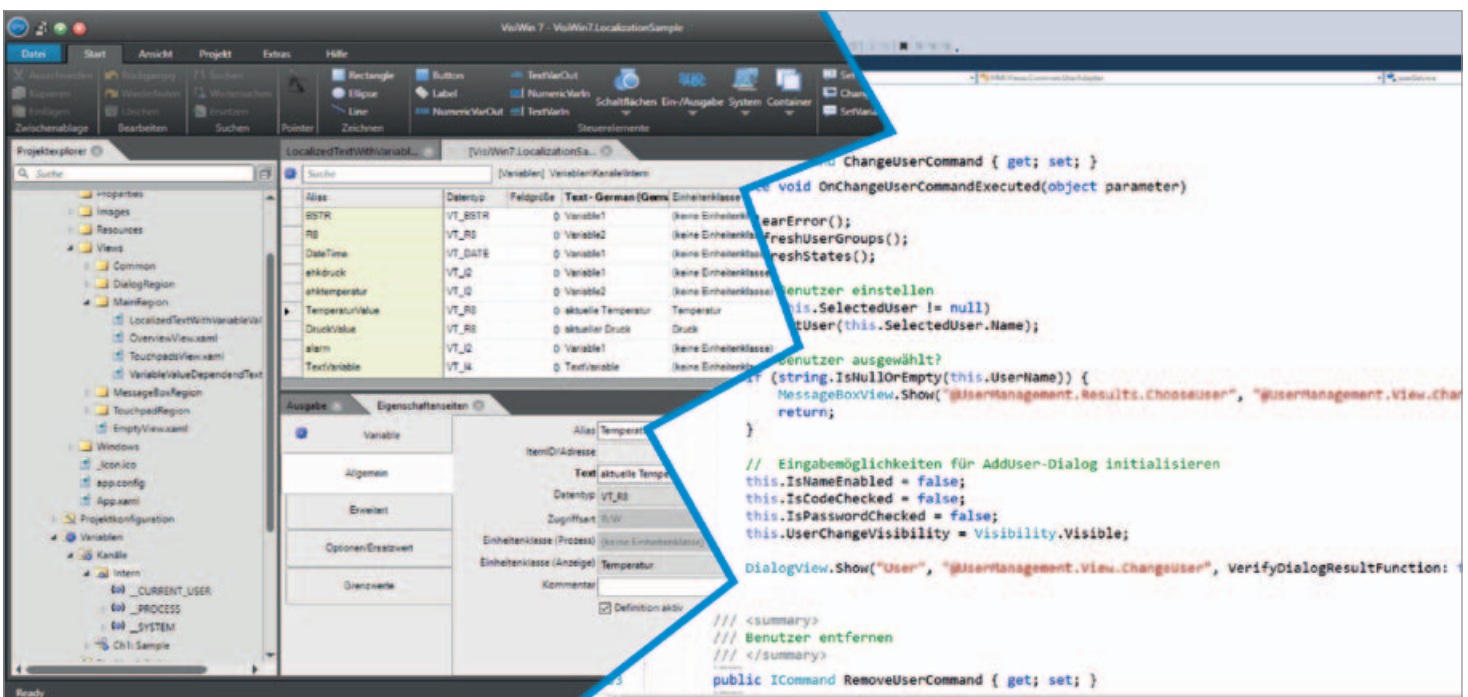


Bild: Inosoft

**Programmieren versus Parametrieren**

Wer selbst programmiert, hat die volle Freiheit, seine Bedienkonzepte und Anwendungsideen umzusetzen – sofern sich der damit verbundene Aufwand wirtschaftlich vertreten lässt. Auch bei der Realisierung stehen sämtliche technischen Türen offen, wenn es etwa um Programmiersprachen oder Entwicklungs-Frameworks geht. Dies ist insofern relevant, als gerade die Schnittstellen für neue Geräte, beispielsweise für Datenbrillen wie Microsofts Hololens, nur in aktuellen Umgebungen zu realisieren sind. Dementsprechend hoch sind die Anforderungen an die Programmierer: Sie müssen fit sein in den aktuellen Programmiersprachen und -Tools.

Wer diesen Aufwand nicht leisten kann oder will, greift zu einer vorgefertigten Lösung, die sich einfach an die eigenen Wünsche anpassen lässt – zumindest im Rahmen der vorgegebenen Funktionen. Ebenso unterliegt man unter Umständen gewissen Einschränkungen beim grafischen Design der Oberfläche. Als Maschinenbauer ist man in diesem Fall darauf angewiesen, dass der Anbieter alle benötigten Optionen realisiert oder entsprechende APIs und Schnittstellen bereitstellt, die eine Ergänzung mit zusätzlichen Produkten oder die Einbindung neuer Bediengeräte und Bedienkonzepte zulassen.

Die zahlreichen auf dem Markt erhältlichen Lösungen unterscheiden sich hier zum Teil gravierend. Hier gilt es genau hinzuschauen, an welchen Hersteller man sich bindet – sowohl in Hinblick auf die aktuelle Technik als auch auf den Entwicklungspfad, der für künftige Versionen gezeichnet wird. Ebenso ist zu prüfen, welche Schnittstellen

Vergleich der Lösungsansätze			
Bewertung * (nachteilig) – ***** (vorteilhaft) – Sterne in Klammern: abhängig von Implementierung, Typ des Systems etc.			
	Parametrierbare Lösung	Programmierte Lösung	Hybridlösung
Initialer Aufwand	****	*	***(*)
Projektbezogener Aufwand	**	***(**)	****(*)
Zielerreichbarkeit	**	**	****(*)
Wartungsaufwand	****	**	***(*)
Verfügbare Zielsysteme	**(***)	*****	****(*)
Abhängigkeit vom Hersteller	*	*****	**(**)
Qualifikation für initiale Erstellung	***	*	**
Qualifikation für Projektierung	***	***(**)	***(**)
Mögliche Effizienzsteigerung Engineering	**	*****	****(*)
Lizenzkosten	*	*****	***
Supportmöglichkeiten	*****	*(*)	*****

zu Anwendungen außerhalb der HMI-/SCADA-Welt eine Lösung unterstützt – etwa zu ERP-Systemen und Cloud-Anwendungen –, und welche Protokolle für den Datenaustausch zur Verfügung stehen. Je nach verfolgter Strategie ist sehr genau zu prüfen, ob mit der vorgesehenen Lösung das Ziel überhaupt zu erreichen ist.

Doch nicht nur die Lösungen selbst unterscheiden sich stark – auch Entwicklung und Pflege haben je nach gewählter Strategie einen völlig anderen Charakter. Wer seine HMI-Anwendung selbst entwickelt, braucht hochqualifizierte Entwickler. Diese sind auf dem Markt nicht leicht zu finden – größere Unternehmen, die attraktive Konditionen bieten können, sind deshalb oft im Vorteil. Nicht zu unterschätzen sind die ‚Nebeneffekte‘ einer selbst pro-

grammierten Anwendung: Anpassung an neue Hardware sowie an aktualisierte Betriebssysteme und Frameworks. Dies kann eine Stärke sein, wenn die selbstentwickelte Lösung die Vorteile der technischen Innovation aufgreifen und daraus Mehrwert für den Nutzer generieren kann. Besondere Aufmerksamkeit verlangt das Versionsmanagement der eigenen Codes sowie der eingesetzten Bibliotheken von dritter Seite. Darüber hinaus muss über die Laufzeit des Systems hinweg die Security gewährleistet sein.

Wer auf eine zugekaufte Lösung setzt, muss sich um solche Probleme nicht kümmern, das ist Sache des Anbieters. Allerdings muss dieser auch verlässlich und zeitnah seinen Pflichten nachkommen. Häufig sind zugekaufte Lösungen jedoch proprietär, also nicht an gängigen offenen Standards orientiert, sondern nutzen herstellerspezifische Daten- und Schnittstellenformate. Damit bindet man sich eng an den HMI-Anbieter, teils nicht nur bei der Software, sondern auch bei der Hardware. Solange der Hersteller sein Produkt regelmäßig an neue Rechner- und Ausgabetechniken anpasst, hat der OEM viel Aufwand gespart. Bleibt das Produkt dagegen auf einer Entwicklungsstufe mit zunehmend veralteter Hardware stehen, kann der Maschinenbauer dies nicht selbst beheben. Im schlimmsten Fall bleibt nur der Wechsel des Anbieters – und damit ein aufwendiger Neubeginn auf der nächsten Plattform.



Die Anforderungen an Bedienkomfort, Funktionalität und Design von HMI-Oberflächen steigen. Andererseits verschaffen hochwertige Oberflächen Maschinenherstellern einen Wettbewerbsvorteil.

## Der Mittelweg

Neben diesen beiden ‚idealtypischen‘ Strategien – Programmieren oder Parametrieren – gibt es einen Mittelweg, der die Vorteile der beiden Welten in sich vereint. Solche Systeme bieten einerseits eine breite Palette von Standardfunktionen, die sich einfach per Parametrierung aufsetzen lassen. Zum anderen lassen sich damit komplexe Funktionen mit Hilfe objektorientierter Programmiersprachen und den dafür verfügbaren professionellen Tools nachrüsten, während vorgefertigte Lösungen allenfalls mit Scriptsprachen in geringem Umfang modifizierbar sind.

Die Idee hinter solchen Lösungen: das ‚Pflichtprogramm‘ der HMI lässt sich mit geringem Aufwand erstellen. Diese Aufgabe können erfahrene Fachkräfte aus der Automation übernehmen. Erst für zusätzliche Erweiterungen kommen dann Software-Experten zum Einsatz. Funktionen, die auf dem Markt nicht als fertige Lösungen erhältlich sind, spezielle Anwendungen und Services des Maschinenbauers sowie die Berücksichtigung kundenspezifischer Technologien und Anforderungen sind nur einige Beispiele für solche Ergänzungen.

Ein wichtiger Aspekt ist auch die Gestaltung der Oberfläche, beispielsweise im Design des Maschinenbauers oder nach Wunsch von dessen Kunden. Entscheidend

dabei ist, dass sich die programmierten Erweiterungen nahtlos in die Teile der HMI einfügen, die auf Parametrieren basiert.

Ein Beispiel für ein solches offenes System ist VisiWin von Inosoft. Diese Lösung ist tief in das Microsoft-Ökosystem zur Entwicklung grafischer Systeme eingebettet. Dazu zählt an erster Stelle Visual Studio, daneben das Designwerkzeug Blend und als Technologie das .Net-Framework inklusive Windows Forms sowie das GUI-Framework WPF (Windows Presentation Foundation). Diese Integration gewährleistet eine kontinuierliche Weiterentwicklung und Anpassung an aktuelle Windows-Betriebssystemversionen und moderne PC-Technik.

Eine große Entwickler-Community und zahlreiche Download-Plattformen bieten sogar außerhalb des weitgehend kostenlosen Hersteller-Services umfangreiche Möglichkeiten, Unterstützung zu bekommen, auch in Form von fertigen Funktionsbausteinen, die zum Download bereitstehen. Die Gefahr eines ‚Hersteller-Lock-In‘ ist bei dieser Lösung also denkbar gering.

Wer selbst Zusatzfunktionen für VisiWin programmieren will, kann auf verschiedene Technologien und Sprachen zurückgreifen. Im Fokus stehen C# und VB.NET (Visual Basic.NET) von Microsoft sowie Web-Technologien wie HTML5, CSS, XAML und Javascript. Selbstverständ-



Offene Schnittstellen ermöglichen die Einbindung von Bedien-geräten der Zukunft – zum Beispiel der Microsoft HoloLens.

lich sind auch aktuelle Kommunikationsprotokolle wie OPC UA integriert.

Zusammenfassend lässt sich festhalten: Jede Lösung hat ihre Berechtigung, jedoch ist jede Anwendung individuell zu betrachten. Grundlagen für eine Entscheidung sind ein gründliches Abwägen von Vor- und Nachteilen sowie der verfügbaren Kompetenzen im Unternehmen. *gh*



**STEFAN NIERMANN**  
ist Vertriebsleiter bei Inosoft.