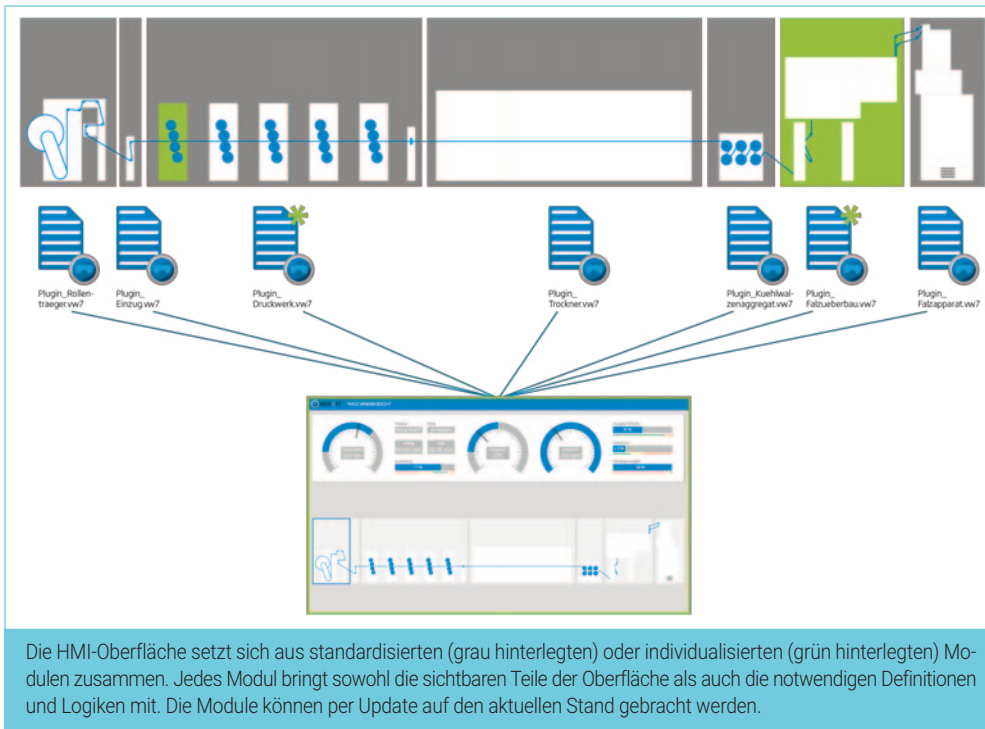


HMI-Erstellung am Beispiel von VisiWin

Modulare HMI für Sondermaschinen

Maschinenbauer modularisieren ihre Produkte, um den Konstruktionsaufwand zu senken und Fehlerquellen abzustellen. Besonders bei der HMI-Software birgt der Ansatz Potential, da die individuelle Erstellung aufwendig und fehleranfällig ist. Doch wie passt die Modularisierung mit den steigenden Individualisierungsanforderungen der Branche zusammen?



werden beispielsweise Korrekturen, die während eines Projekts am HMI-Gerüst vorgenommen werden, nicht in die Vorlage übertragen. Neue HMI-Projekte bergen damit den gleichen Fehler immer wieder.

In Modulen standardisiert

Wie kann eine Lösung aussehen? Grundsätzlich ist eine Bibliothek, die zentral gepflegt und in den Projekten verwendet wird, ein möglicher Weg. Doch dabei kommt man um einen möglichst hohen Grad der Standardisierung nicht herum. Wie eingangs erwähnt, soll es hier aber um Sondermaschinen gehen, bei denen viel individualisiert wird. In diesem Fall sollte zunächst der komplette Rahmen der HMI-Applikation standardisiert sein. Dafür sprechen schon die Wiedererkennbarkeit des User Interfaces und die Durchgängigkeit bei der Maschinenbedienung. Dieser Rahmen kann sehr weit gefasst sein und neben dem Hauptfenster mit Seitenaufteilung und Navigation auch grundsätzliche Einstellseiten, die Sprachumschaltung, das User-Management, Seiten für Störmeldungen inklusive Historie, Seiten für Trendcharts, die allgemeine Rezepturverwaltung und die Anzeige des Maschinen-Logbuchs enthalten. Denkt man weiter, kommen einem Im- und Export von Daten, Erzeugung von Reports und viele weitere Funktionen in den Sinn. Bei jedem Punkte lohnt es sich, über einen generischen Ansatz nachzudenken und diesen möglichst auch umzusetzen. Übrig bleibt der individuelle Anteil, der schon nicht mehr so groß ist. Bei genauer Betrachtung gibt es weitere Details, die sich standardisieren lassen.

O bwohl viele Maschinenbauer individuell auf die Bedürfnisse ihrer Kunden abgestimmte 'Unikate' fertigen, weisen die verschiedenen Maschinenkonfigurationen oft gleiche oder ähnliche Baugruppen auf. Um die Human Machine Interfaces (HMI) in der Entwicklung nicht jedes Mal neu erstellen zu müssen, wird oft ein existierendes Projekt als Vorlage genommen, dupliziert und angepasst. Entstehen HMI-Applikationen nach diesem Schema, führt das zu einer unüberschaubaren Versionsvielfalt und die einheitliche Pflege der Software wird irgendwann praktisch unmöglich.

Rumpfmachine als Basis?

Ein anderer Ansatz erspart ebenfalls Fleißarbeit und erhöht zudem die Durchgängigkeit: Es wird ein Projekt als Kopiervorlage erstellt, welches alle Basisfunktionen sowie die benötigten Komponenten und technischen Möglichkeiten beinhaltet. Jedes neue Projekt wird aus dieser Vorlage kopiert. Nicht benötigte Teile werden entfernt und die individuellen Teile dann je nach Maschine erstellt. In beiden Fällen liegt also bereits ein gewisser Standard zugrunde. Doch die Pflege der Standards fällt aus Zeitgründen oft unter den Tisch. Dann

Etwa Kombinationen aus Aktoren und Sensoren, die immer gleich verbaut werden, sollten auch in der Oberfläche des HMI immer gleich sein. Oft gibt es bei diesen Anzeigeelementen zugehörige Dialoge, die weitere Bedienmöglichkeiten anbieten. Das können z.B. Rezeptwerte oder Handfunktionen sein. Auch hier ist trotz der notwendigen Individualisierung meist ein hoher Prozentsatz an standardisierbaren Elementen zu finden.

Modularisierung mit VisiWin

Alle standardisierten Module gilt es im nächsten Schritt in einer zentral gepflegten Bibliothek abzulegen. Diese Module sollten nicht als bearbeitbare Komponenten im Projekt liegen, damit deren Pflege nur an zentraler Stelle erfolgen kann. Modularisierung muss in diesem Bereich eine Einbahnstraße sein, sonst funktioniert das Konzept nicht. In VisiWin 7 von Inosoft gibt es dafür einen Projekttyp, der diese Form der Modularisierung unterstützt. Den Rah-

men für Projekt bildet die sogenannte Shell. Sie enthält das Hauptfenster der Anwendung mit den Basisfunktionen. Alle Funktionen, die man hier unterbringen sollte, finden sich im oben beschriebenen Ansatz zur Modularisierung. Die weiteren Funktionsmodule heißen in der HMI-Software Plugins. Diese enthalten Bildschirmseiten und Logiken inklusive ihrer Definitionen (Variablen, Texte, Störmeldungen, Rezeptvariablen usw.). Sie sind als Teilprojekte zu verstehen, denen nur der Anwendungsrahmen fehlt. Den bekommen sie mit der Shell. Ein weiterer Ansatz für Module sind kleinere Bildelemente, die in VisiWin, wie in vielen Hochsprachenumgebungen, User-Controls genannt werden. Sie enthalten mehrere Unterelemente und werden meist an eine Struktur (User Defined Type) im Steuerungsprogramm gebunden. Die Unterelemente verbinden sich mit den Mitgliedern der Struktur und zeigen deren Zustand an. Ein Beispiel kann eine Klappe mit mehreren Sensoren und unterschiedlichen Zuständen sein.

Praxisbeispiel Druckmaschine

Ein Beispiel für ein in größeren Teilen modularisiertes Projekt ist das HMI einer Druckmaschine (siehe Bild). Jedes ihrer Teile kann als Modul abgebildet werden. Für die Druckwerke lassen sich die Module noch für jede Druckfarbe als Instanz des Moduls anlegen. Alle Module sollten sinnvoll zusammengefasst in Bibliotheken zur Verfügung stehen. Die Verwaltung der Bibliotheken sollte möglichst einem geeigneten Tool übertragen werden, das Versionierung unterstützt und parallele Entwicklungszweige erlaubt. All dies zusammengefasst ergibt einen HMI-Baukasten, aus dem in einer Mischung von standardisierten und individuellen Bausteinen zügig und in hoher Qualität Applikationen entstehen.

Stefan Niermann
 Head of Business Development
 Inosoft GmbH
www.inosoft.com