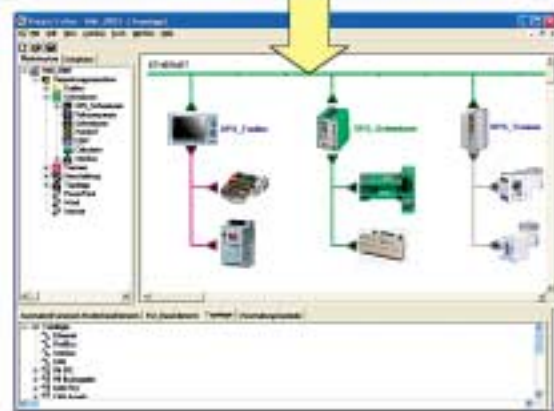


Boris Waldeck

Vom Modul zum lauffähigen Gerät

Sind für die einzelnen Module einer Maschine die Softwareteile und Konfigurationsdaten vollständig, fehlt letztlich noch die Verknüpfung der Module zur Gesamtanlage: Mit den Netzwerk-, Topologie- und Verschaltungseditoren stellt das Automation Framework die hierfür notwendigen Werkzeuge bereit.



Wie wird aus einzelnen Modulen eine lauffähige Maschine? Nehmen wir das Beispiel einer Maschine, die sich aus den Funktionen Füllen, Schweißen und Trennen zusammensetzt:

Vorausgesetzt, die Softwareteile und Konfigurationsdaten der einzelnen Automatisierungsmodule sind vorhanden, so gilt es noch, die Netzwerkverbindungen und Verschaltungen aller Module für die Gerätekonfiguration festzulegen.

Im Netzwerkkeditor erfolgt zunächst die Festlegung und Darstellung, welche Ge-

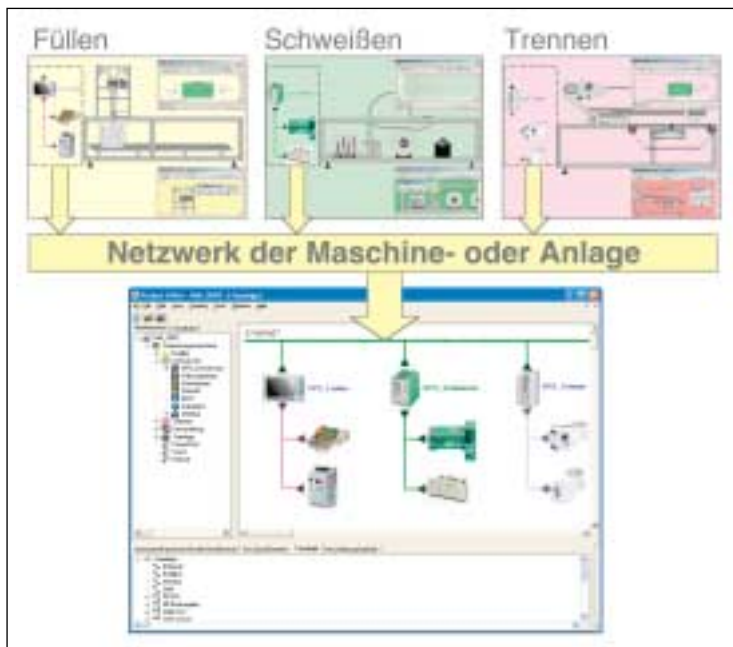
räte an welchem Kommunikationsnetzwerk anzuschließen sind. In unserem Beispiel liegen für die Automatisierungsmodule Füllen, Schweißen und Trennen drei Feldbusse mit je einem Master vor. Diese Mastergeräte sind auf der oberen Ebene über ein Ethernet-Netzwerk miteinander gekoppelt.

Wäre kein überlagertes Netzwerk vorhanden, so könnten auch alle Komponenten an einem Netzwerk hängen (zum Beispiel Profibus). Jedes Modul würde dann einen Teilstrang des Gesamtnetz-

werkes der Maschine darstellen. Neben dem Netzwerk-Editor werden mit dem Topologie-Editor die realen physikalischen Geräte und ihre Verbindungen dargestellt. Der Editor stellt zum Beispiel dar, welche Arten von Kabel verwendet werden und ob es sich bei dem Netzwerk um eine Linienstruktur oder um eine Ringstruktur handelt.

Über die Festlegung der Verbindung zwischen den Geräten hinaus, ist es notwendig, zu definieren, welche Daten zwischen den Geräten ausgetauscht werden

(Grafik: Computer & AUTOMATION, Quelle: Lenze, KW)



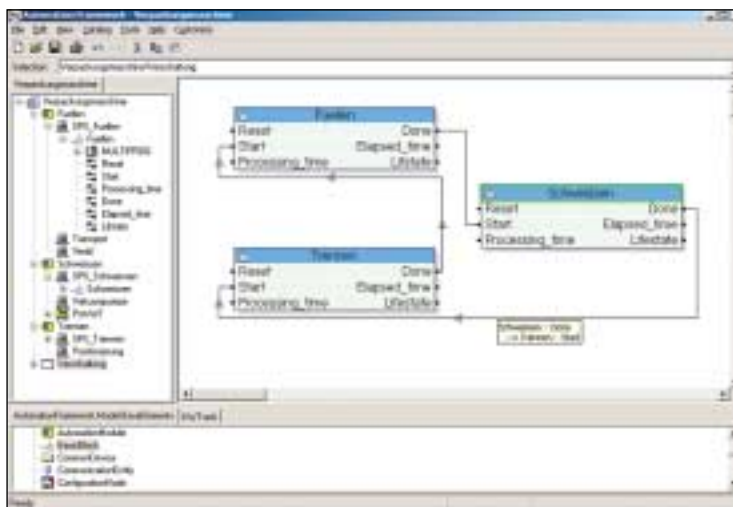
Das Netzwerk der Maschine mit den Funktionen Füllen, Schweißen und Trennen, dargestellt im Netzwerkeditor.

sollen und wie die Ergebnisse aus der Logikarbeitung eines Automatisierungsmoduls von anderen Modulen weiterverarbeitet werden. Dies geschieht mittels Verschaltungseitor. Eine solche Verschaltung wird dabei nicht zwingend auf mehreren Geräten verteilt abgearbeitet. Es ist durchaus möglich, dass die Applikationslogik aller drei Automatisierungsmodule auf einem zentralen Gerät verarbeitet wird.

Ist die Definition von Netzwerk, Topologie und Verschaltung erfolgt, sind alle notwendigen Informationen für die Konfiguration der Kommunikation vorhanden. In der offenen Systemplattform Automation Framework sind alle drei Editoren konfigurierbar, so dass sie auf beliebige Netzwerke angepasst werden können.

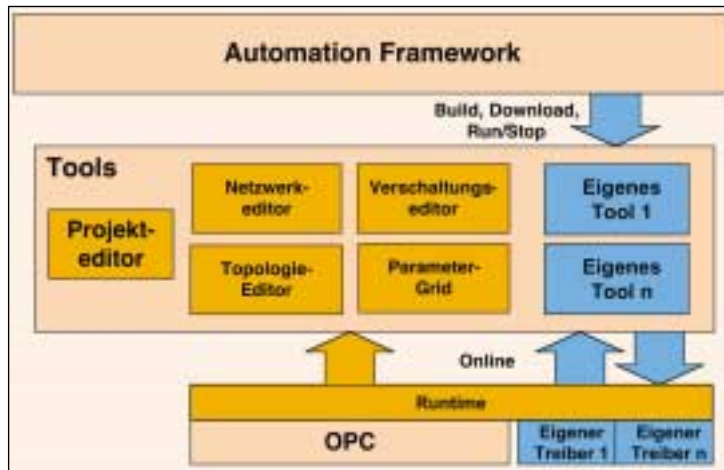
Der Download

Mittels der Build-Funktion des Automation Framework erfolgt der Anstoß zur Erzeugung der Gerätekonfigurationen und Geräteprogramme. Dies kann für einzelne Geräte bis hin zu einem



Der Verschaltungseitor: Die logische Abarbeitung und Verkettung der Module.

Das Geräte-konzept: Im Automation Framework werden Geräte über eine offene Runtime-Schnittstelle eingebunden.



ganzen Projekt erfolgen. Treten beim Build Fehler auf, so werden diese im allgemeinen Meldungsfenster von Automation Framework eingetragen. Von hier aus lassen sich die verschiedenen Fehlerstellen per Doppelklick näher analysieren.

Nach dem Build lassen sich die Konfigurationen und Programme auf alle Geräte herunterladen. Der Anwender kann dies wiederum von zentraler Stelle aus starten. Der Automation Framework sorgt sodann dafür, dass alle Tools ein Download an ihre zugehörigen Geräte vornehmen. Parameter, Programme, Netzwerkkonfigurationen und alle weiteren änderbaren und zur Laufzeit nötigen Informationen werden so auf die Geräte

übertragen. Über Run und Stop können in Automation Framework Teile oder aber auch die ganze Maschine gestartet und gestoppt werden. Sofern ein Gerät dies unterstützt, lassen sich darüber hinaus die Projektdaten mit auf dem Gerät ablegen. Automation Framework fasst hierfür alle entsprechenden Projektdaten in einer zip-Datei zusammen.

Geräte-Informationen online

Neben dem Offline-Modus für das Engineering, stellt Automation Framework im Online-Modus aktuelle Geräte-werte dar. Dies können aktuelle Parameter aber auch Variablenwerte von Programmen sein. Sofern für die Geräte ein OPC-

Server zur Verfügung steht, kann das Lesen der Onlinewerte über diese OPC-Anbindung erfolgen. Neben der OPC-Schnittstelle ist eine konfigurierbare Runtime-Schnittstelle zur Implementierung gerätespezifischer Kommunikation vorhanden.

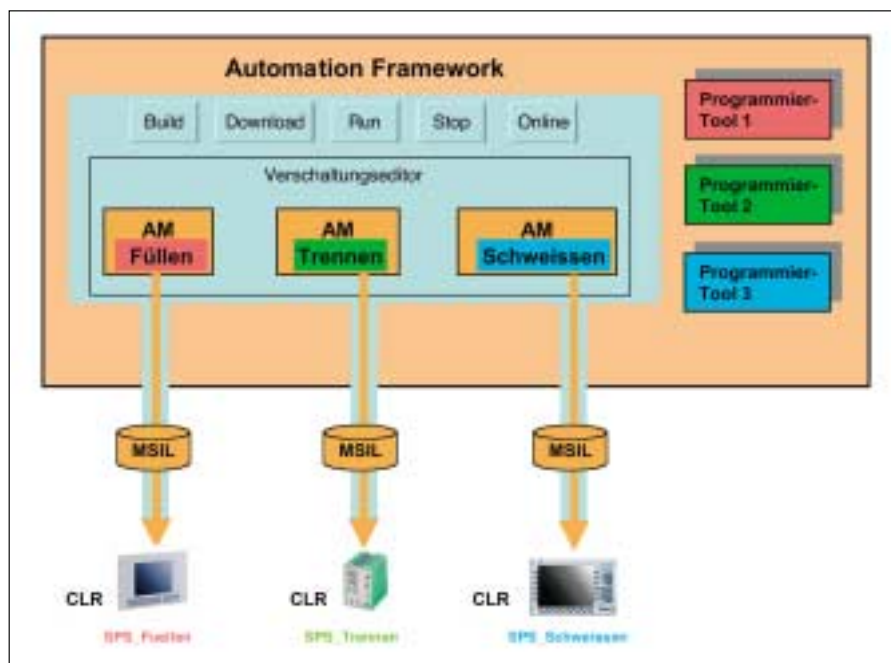
.net – Impuls für eine Geräte-Unabhängigkeit

Die Vorteile eines integrierten Engineering mit Automation Framework kommen besonders zum Tragen, wenn mehrere Steuerungen in einer Automatisierungsanlage zusammenarbeiten. Dies funktioniert auch, wenn Steuerungen unterschiedlicher Hersteller eingebunden sind. In diesem Falle liefert der Hersteller zu seinem Gerät ein Programmier- oder Konfigurationstool, mit dem sich die gerätespezifischen Daten für das jeweilige Gerät generieren und laden lassen. Nachteil bislang ist allerdings, dass durch dieses Verfahren die Wiederverwendung der Automatisierungsmodule an die jeweils verwendete Hardware gebunden ist.

Microsoft .net bietet durch das Konzept des Plattform-unabhängigen Zwischencodes MSIL und durch das CLR-Konzept eine Perspektive, um hier neue Wege zu gehen.

In .net steht CLR für Common Language Runtime. Es ist das einheitliche Laufzeitsystem für verschiedene Programmiersprachen, Betriebssysteme und Prozessoren. Die CLR wird mit der MSIL (Microsoft intermediate language) programmiert. Die Compiler für C++, C# oder Visual-Basic erzeugen also keinen processorabhängigen Objektcode mehr, sondern MSIL-Zwischencode, der von der CLR in den Geräten abgearbeitet werden kann. Wird dieses Konzept auch für Geräte in der Automation genutzt, stehen im Automation Framework geräteunabhängige Automatisierungsmodule zur Verfügung. Die Wiederverwendung solcher Module wird somit noch einmal stark vereinfacht.

hap
(Siehe hierzu auch Computer&AUTOMATION, Heft 11/2003)



Erfolgt die Anbindung der Steuerungen zukünftig nicht mehr über eine gerätespezifische Kommunikation, Code oder Daten, sondern über die .net-basierte CLR, bringt dies einen gewaltigen Schritt in Richtung geräte-unabhängiger Automatisierungsmodule.

(Bilder: KW)



Boris Waldeck

ist Manager Marketing und Vertrieb bei der KW-Software GmbH.