



Beispiel für ein Smart Object Template mit dem Motor-Symbol und seinen freigegebenen Eigenschaften.

SERIE: EFFIZIENTES PROJEKTIEREN MIT ZENON
TEIL 2

WIE SIE PROJEKTE MIT ZENON ZENTRAL MANAGEN

IU

Erschienen in
INFORMATION UNLIMITED
DAS MAGAZIN VON COPA-DATA
Nr. 36, November 2020
© Ing. Punzenberger COPA-DATA GmbH
www.copadata.com/iu

Im ersten Teil dieser Serie ging es um die grundlegende Philosophie von zenon und die darauf basierenden Konzepte. Thema war unter anderem die zentrale Verwaltung von Elementen. Jetzt erfahren Sie anhand eines praxisnahen Beispiels, wie die zentrale Verwaltung mithilfe von Datentypen, Symbolen, Verknüpfungen und – mit der neuen Version zenon 8.20 – Smart Objects und Smart Object Templates optimiert wird.

Stellen Sie sich vor, Sie möchten in Ihrem zenon Projekt drei Motoren desselben Typs für Förderbänder in einer Produktion visualisieren. Diese Motoren werden per HMI ein- und ausgeschaltet und liefern dem System Informationen über Betriebsstatus, Drehzahl und Temperatur. Um die Motoren in zenon abzubilden, benötigen Sie Variablen, Funktionen und ein Bild oder mehrere. Hier kommen die zenon Datentypen und Symbole ins Spiel, denn sie bieten Ihnen viele Möglichkeiten bei der Projektierung.

WAS SIND DATENTYPEN?

Datentypen erlauben Ihnen die zentrale Definition der Eigenschaften von Variablen, wie beispielsweise Messbereiche oder Grenzwerte. Im Beispiel legen Sie mehrere Einfache Datentypen an: „MotorTemperature“ (BYTE), „MotorState“ (BOOL), „MotorSpeed“ (UINT). Bei „MotorTemperature“ tragen Sie als Maßeinheit „°C“ ein und erstellen einen Grenzwert für „kritisch“ ab 80 °C und „Überhitzung“ ab 100 °C. Anschließend legen Sie einen Struktur-Datentyp „Motor“ mit den Struktur-Elementen „Temperature“, „Speed“ und „State“ an, welche jeweils auf den zuvor erstellten Einfachen Datentypen basieren. Achten Sie darauf, dass unter „Struktur-Optionen“ „Datentyp verknüpfen“ gewählt ist, damit sich später am Datentyp durchgeführte Änderungen automatisch auf die Struktur-Elemente auswirken.

Im nächsten Schritt legen Sie nun die Variablen für die drei Motoren mit Namen „Motor_Band1“ (..2, ..3) an, die jeweils auf dem Datentyp „Motor“ basieren. Insgesamt wurden jetzt automatisch 9 Variablen mit Ihren Vorgaben erstellt und aktiviert.

SYMBOLE SINNVOLL VERWENDEN

Zur grafischen Visualisierung des Motors nutzen Sie als zentrale Definition ein Symbol. In diesem Fall empfiehlt sich ein Icon mit Beschriftung und Wertanzeigen für Temperatur und Drehzahl, die Sie jeweils mit den entsprechenden Variablen aus dem Struktur-Datentyp „Motor_Band1“ verknüpfen.

Ziehen Sie das eben angelegte Symbol in ein Bild, wo es für „Motor_Band1“ direkt verwendet werden kann. Für

„Motor_Band2“ ziehen Sie das Symbol ein weiteres Mal ins Bild und tauschen in dem automatisch geöffneten Dialog die Variablen und Funktionen zusammen per Verknüpfungsregel aus. Alternativ können Sie diesen Dialog auch im Eigenschaftenfenster unter Verknüpfungsregel öffnen. Im Beispiel ist für die Ersetzungsfunktion als Quelle „*1“ und als Ziel „2“ ausreichend. Über die Schaltfläche „Vorschau“ können Sie das Ergebnis dieser Verknüpfungsregel sofort überprüfen.

SYMBOLEIGENSCHAFTEN INDIVIDUALISIEREN

Eigenschaften wie beispielsweise die Farbe oder der Text des Motors werden von der Verknüpfungsregel jedoch nicht erfasst. Damit diese Eigenschaften direkt im Bild und bei jeder Verwendung eines Symbols individuell anpassbar sind, können Sie einzelne Eigenschaften von Elementen innerhalb eines Symbols freigeben.

Dazu öffnen Sie das Symbol, markieren das Element und geben die gewünschte Eigenschaft (z. B. Text) mit einer dieser drei Möglichkeiten frei, wobei Sie immer von der Beschriftung der Eigenschaft im Eigenschaftenfenster ausgehen müssen:

- die Eigenschaft in den Bereich unterhalb der Zeichenfläche des Symbols ziehen
- per Rechtsklick das Kontextmenü öffnen und „Text freigeben“ wählen
- die Eigenschaft markieren und in der Symbolleiste „Eigenschaft freigeben“ wählen

Selektieren Sie das Symbol nun im Bild, sehen Sie im Eigenschaftenfenster einen neuen Knoten in der Baumansicht „\$_<Elementname>“, der die eben freigegebene Eigenschaft enthält. Das erlaubt Ihnen, bei jeder Symbolverwendung individuelle Definitionen lokal zu überschreiben. Gleichzeitig vermindert es die Komplexität für den Engineer, da nur die für ihn relevanten Eigenschaften angezeigt werden.

Möchten Sie den Motor direkt aus diesem Symbol heraus ein- bzw. ausschalten, müssen Sie zwei Buttons und

zwei Sollwert-setzen-Funktionen mit Namen „Motor_Band1.Start“ beziehungsweise „Stop“ auf der Variable „Motor_Band1.State“ anlegen. Nach Speichern des Symbols werden automatisch alle Verwendungsstellen im Bild aktualisiert. Öffnen Sie jetzt für „Motor_Band2“ erneut den Dialog für die Verknüpfungsregel, dann sehen Sie in der Zielspalte, dass für die beiden Funktionen Start und Stop ein „(?)“ am Ende steht. Das bedeutet, diese Funktionen existieren nicht im Projekt. Als Verwender dieses Symbols müssen Sie also wissen, welche Variablen und Funktionen vom Symbol erwartet werden, diese im Projekt entsprechend anlegen und die Funktionsparameter anpassen.

DAS MUSS DOCH EINFACHER GEHEN!

Mit der aktuellen Version von zenon 8.20 wird vieles leichter. Sie können nun Smart Objects und Smart Object Templates anlegen. Diese fassen zusätzlich zur grafischen Darstellung von Symbolen und Bildern auch Datentypen, Reaktionsmatrizen, Funktionen, Verriegelungen, Dateien und weiteres zusammen.

Dafür gibt es im zenon Editor einen eigenen Bereich für die Smart Object Templates. Legen Sie im linken Bereich ein neues Template „Band“ an. Im rechten Teil des Fensters erscheint nun ein Baum mit den verfügbaren zenon Modulen. Hier legen Sie dieselben Objekte wie oben beschrieben an, können jedoch an allen Stellen das „Band1“ im Namen weglassen.

Im Symbol geben Sie die freigegebene Eigenschaft „Label“ auch für das Smart Object Template frei, indem Sie die Eigenschaft im Symboleditor markieren und über das Icon in der Toolbar erneut freigeben. Nun wechseln Sie zurück zum Projektbaum, wählen dort den Knoten Smart Objects aus und erstellen ein neues Smart Object, das auf dem eben erstellten Template basiert. In der Liste sehen Sie das Smart Object und darunter das von Ihnen definierte Symbol. Ziehen Sie dieses Symbol nun in ein Bild und öffnen anschließend erneut den Dialog für die Verknüpfungsregel, ist bereits die korrekte Regel für die Ersetzung eingesetzt, und in der Vorschau werden alle Ziele gefunden.

Wenn Sie also ein Smart Object erzeugen, werden automatisch alle projektierten Variablen, Funktionen etc. im Projekt angelegt und bei allen Verwendungsstellen angepasst.

SO ERWEITERN SIE SMART OBJECT TEMPLATES FÜR VIELE ZWECKE

Benötigen Sie in Ihrer Anlage auch in anderen Baugruppen Motoren, können Sie das Smart Object Template auch in anderen Smart Object Templates als Referenz hinzufügen und so beliebig oft verwenden. Und möchten Sie später den Motor zum Beispiel um eine Detailansicht mit Trendkurven erweitern, ist das mit Smart Object Templates ebenfalls

problemlos möglich. Legen Sie dazu ein neues Bild an und projektieren Sie je ein Trend-Element für „Motor.Speed“ und „Motor.Temperature“ und zusätzlich einen Button mit der Funktion Schablone schließen. Sobald Sie das Smart Object Template speichern, werden alle Smart Objects im Projekt aktualisiert sowie die dazu notwendigen Funktionen und Bilder angelegt. Die neue Funktionalität steht sofort im gesamten Projekt zur Verfügung.

JETZT DIE VORTEILE DER NEUEN VERSION ZENON 8.20 NUTZEN

Das Konzept der Verknüpfung, das Sie oben bei den Symbolen kennengelernt haben, steht auch in der Funktion Bildumschaltung zur Verfügung. Bei Smart Objects wird sie beim Erzeugen automatisch angepasst, sodass im Detailbild immer die richtigen Variablen verknüpft sind.

Sie haben in diesem Teil der Serie erfahren, wie Sie auf Basis der zenon Philosophie wiederverwendbare Komponenten mithilfe von Datentypen, Symbolen, Verknüpfung und mit der neuen Version auch Smart Objects anlegen und beliebig oft wiederverwenden können. Im nächsten Teil der Serie werden Sie die Details der Verknüpfung genauer kennenlernen.



GERO GRUBER
Product Manager

Als Product Manager und Product Owner für die zenon Softwareplattform liegt sein besonderer Fokus auf dem User Interface, dem Interaktionsdesign für die gesamte Plattform sowie der grafischen Visualisierung in der zenon Runtime.