

ComNorm – Parametrisierung, Version 1.4

Parameterisierung

Hersteller und Händler bieten oft ein Produkt in verschiedenen Ausführungen an. Sind dies nur wenige Variationen, so lässt sich für jede Konfiguration eine separate Bestellnummer definieren. Gibt es jedoch mehrere zu bestimmende Konfigurationsparameter, so ist die Abdeckung mit einem abgeschlossenen Nummernsystem nicht mehr möglich.

Beispiel:

Eine Firma stellt Korpusmöbel her. Der Kunde kann unter folgenden Vorgaben auswählen:
5 Höhen, 6 Breiten, 3 Fussarten, 1..10 Tablare, 5 Oberflächen, 3 Frontarten, 10 Griffen und 18 Farben.
Dies alleine ergibt 2'430'000 mögliche Kombinationen und ebenso viele Nummern.

Um dieser Problematik entgegenzutreten, wird die Verwendung von Parameter unterstützt. Das heisst, um ein Bestell-Produkt genau zu spezifizieren, kann neben der Bestellnummer ein zusätzlicher Parameterstring angegeben werden. In diesen lassen sich beliebig viele Parameterwerte eincodieren.

Ideen von Parametrisierungsstrings:

```
"Length=1200,Height=700,Width=800,Color=gray"  
"L=1250|C=red|SP=3434.44|T=63R5"  
"34534EETSE34TZTRZ456234"
```

In der Praxis gibt es fast für jedes Produkt eine andere 'beste Art', wie die Erfassung der produktspezifischen Parameter erfolgen soll. Aus diesem Grund wird bei ComNorm nur definiert, wie der Parameterstring ausgetauscht wird, nicht aber was im Parameterstring enthalten ist. Das Konfigurieren geschieht im Browser in einer vom Anbieter gelieferten Seite. So ist es möglich, dass z.B. für das Bestimmen der Produktfarbe ein Lieferant eine Combobox verwendet, und ein anderer, farbige Flächen zum daraufklicken.

Die Verwendung der Parametrisierung muss von den Programmen auf Anbieterseite und der Anwenderseite unterstützt werden und funktioniert wie folgt:

Schritt 1: Serverseite gibt URL der Konfigurationsseite zurück

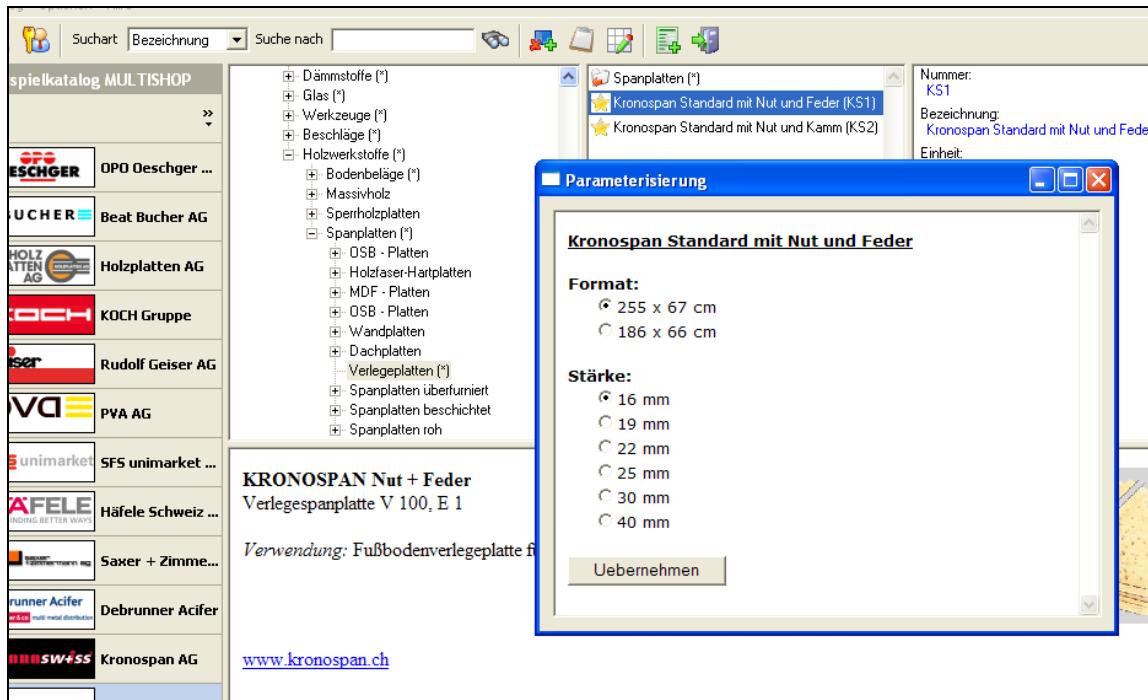
Auf die Anfrage 'GetProductInfo' wird in der Antwort unter dem Element 'ConfigurationData' angegeben, welches die Internetseite ist, die aufgerufen werden muss, um die Parametrisierung vorzunehmen. Ebenso werden mit dem darin vorkommenden Element 'ParamData' die Standardwerte festgelegt. Wie die Eincodierung der Parameterwerte im Element 'ParamData' erfolgt, ist dem Anbieter überlassen.

```
...  
<ProductInfo>  
    <ProductNumber>43.139.44</ProductNumber>  
    ...  
    <ConfigurationData>  
        <ConfigurationURL>http://www.katalog.ch/config.exe?P=5678</ConfigurationURL>  
        <ParamData>D16/W100</ParamData>  
    </ConfigurationData>  
    ...  
</ProductInfo>  
...
```

Schritt 2: Anwenderseite ruft Konfigurationsseite mit Zusatzparameter auf

Wünscht der Anwender die Konfiguration eines Produktes vorzunehmen, so ruft er via die Anwendung die Konfigurations-Internetadresse auf. Vor dem Aufruf erweitert die Anwendung die URL mit einem Zusatzparameter 'ResponseURL' der eine Adresse des lokalen Systems enthält (Siehe dazu weiter unten).

Sobald die gewünschte Seite angezeigt wird kann der Anwender darin die Parametrisierung vornehmen.



Beispiel:

Der erweiterte Aufruf könnte etwa so aussehen:

`http://www.katalog.ch/config.exe?P=5678&ResponseUrl=http://127.0.0.1/abc`

Schritt 3: Serverseite baut Konfigurationsseite nach Vorgaben auf

Auf Anfrage für die Konfigurationsseite gibt der Server die entsprechende statische oder dynamisch aufbereitete Seite zurück. Wichtig ist, dass diese Seite so gestaltet ist, dass mit dem Akzeptieren der Angaben, eine Rückmeldung an die Clientanwendung möglich ist.

Die Rückmeldung erfolgt, indem die mit 'ResponseURL' erhaltene Adresse aufgerufen wird. Die Parametrisierungsinformationen werden als Form-Codierter Parameter 'ParamData' mitgegeben und mit dem Post-Mechanismus übertragen.

Beispiel einer Konfigurationsseite:

```
<html>
    <head>
        <script language="Javascript">
            function getParamData(form) {
                ... Parameter werden in einen String eincodiert ...
                return(pData);
            }
        </script>
    </head>
    <body>
        <form action="http://127.0.0.1/abc" method="post"
              name="ResponseForm">
            ...
            ...
        </form>
    </body>
</html>
```

```
<INPUT TYPE = "hidden" name="ParamData" value="">
<INPUT TYPE = "button" VALUE = "Übernehmen"
       onClick="ParamData.value=getParamData(...);submit();">
</form>
</body>
</html>
```

Schritt 4: Anwenderseite:

Die Clientanwendung setzt einen Filter auf den entsprechenden Socket (oder verwendet eine geeignete Softwarekomponente) und fängt den Aufruf ab bevor die URL angezeigt wird. Sie analysiert die Parameter und aktualisiert die Produktinformationen mit einer Serveranfrage vom Typ 'GetProductInfo'. Dabei gibt sie die Parametrisierungsinformationen mit, so dass der Server die Daten korrekt aufbereiten kann.

Beispiel:

Im obigen Formular wäre der Aufruf der mit dem Drücken der Schaltfläche Übernehmen ausgelöst würde etwa:

http://127.0.0.1/abc

und im Body wäre

ParamData={hier_sind_die_codierten_konfigurationsdaten}

Die Anwendung fängt diesen Aufruf ab und extrahiert den Wert des Parameters ParamData. Zusammen mit der Produktnummer (ProductNumber) ist dadurch die Konfiguration eindeutig bestimmt.

Vermerk zum Parametrisierungsstring

Eine Bekanntgabe des Zusammenbaus und Inhaltes des Strings ist nicht erforderlich, solange die Konfiguration über die vom Anbieter zur Verfügung gestellte Konfigurationsseite erfolgt. Gewisse Softwarehäuser haben aber unter Umständen ganz spezielle Branchenlösungen, bei denen eine ausgewählte Produktpalette des Anbieters direkt in die Anwendung integriert ist. In diesem Fall könnte evtl. eine Struktur- und Inhaltsbeschreibung von Nutzen sein, um den Softwarehersteller eine optimale Integration in die Lösung zu ermöglichen.

Änderungen, History

| Änderungen gegenüber Vorgängerversionen | Version |
|---|---------|
| Erste Dokumentversion | 1.2 |
| Anpassungen | 1.3 |
| | |