



Die Reportingkomponente List & Label in der Praxis

Wie gedruckt

Berichtskomponenten haben sich in den vergangenen Jahren zu mächtigen Tools entwickelt. Sie sorgen nicht nur für die druckreife Ausgabe von Daten, sondern bieten inzwischen vollständige Entwurfsumgebungen auch für den Endanwender. List & Label von combit lässt der jeweiligen Anwendung dabei alle nötigen Freiheiten für eigene Anpassungen.

In puncto Druckausgabe behelfen sich viele Entwickler bei ihren Kundenprojekten selbst. So entstehen Lösungen, die für die jeweiligen Aufgaben zunächst passend erscheinen. Doch dann will der Kunde mehr, und schon beginnt der Marsch durch die Hölle aus Anpassungen des vorhandenen Mechanismus und „schmutzigen“ Tricks für verschiedene Office-Versionen. Irgendwann ist dann doch klar: Eine saubere Lösung muss her.

Neben diesen „klassischen“ Druckmöglichkeiten bieten moderne Reporting-Tools in der Regel mächtige Designer, die auch vom Endanwender zu bedienen sind. Zudem besteht die Möglichkeit, das gedruckte Werk in einer Vielzahl unterschiedlicher Formate auszugeben. Neben PDF stehen hier RTE, HTML, XML und viele weitere Formate zur Verfügung. Die Codebasis für die Berichte ist dabei stets die gleiche. Was der Anwender im Designer gestaltet, ist für die Anwendung völlig transparent.

Zu diesen Tools gehört auch List & Label von combit. Die Steuerung der Berichtskomponente kann auf verschiedene Arten geschehen. Möglich ist die bequeme Variante der Datenbindung, die Anwendung kann aber Datensteuerung und Druckablauf auch komplett selbst in die Hand nehmen. Dies kann an jeder beliebigen Stelle des automatischen Druckablaufs erfolgen; Anpassungen des Standardverhaltens sind somit leicht in jeder Beziehung möglich. Aber auch im Fall der Datenbindung verfügt die Anwendung aufgrund einer Vielzahl spezieller Events noch über Möglichkeiten, das Ergebnis zu verändern.

Den einfachsten Fall zeigt das folgende Listing:

```
ListLabel LL = new ListLabel();
LL.SetDataBinding(dataSet, "");
LL.Design();
```

Grundlage ist ein *ListLabel*-Objekt. Schon dieser einfache Code gibt alle vorhandenen Tabellen und Beziehungen des übergebenen *DataSet* an List & Label weiter. Im Designer, der in Abbildung 1 zu sehen ist, lassen sich somit hierarchische Berichte über mehrere Ebenen erstellen.

Der Druck funktioniert nach dem gleichen Prinzip. Statt der *Design()*-Methode wird dann der folgende Aufruf verwendet:

```
LL.Print()
```

In der Praxis bewährt

Soweit stellt sich die Implementierung zumindest theoretisch sehr einfach dar. Wie aber ist es mit Anwendungen aus dem „richtigen Leben“? Es stellt sich die Frage, wie groß der Aufwand ist, List & Label in eine bestehende Anwendung zu integrieren, und wie lange es dauert, bis sich dieser Aufwand lohnt.

Das Unternehmen Holometric Technologies beispielsweise entwickelt Software

und bietet Dienstleistungen zur Digitalisierung, Visualisierung und Bearbeitung von komplexen dreidimensionalen Körpern und Freiformen für CAD. Im April 2007 kommt die neue Version des AF Application Framework auf den Markt, in das List & Label eingebunden ist; AF Application Framework ist eine modulare Anwendung für 3D-Messtechnik.

List & Label übernimmt in diesem Produkt die Aufgabe, aus Messergebnissen Berichte zu erzeugen. Dabei passt es sich so nahtlos in die Applikation ein, dass der Kunde nicht bemerkt, dass eine externe Komponente im Spiel ist, siehe Abbildung 2.

Die komplexen Visualisierungen der Daten erfolgen dabei direkt in der Anwendung, die das Ergebnis dann einfach an die Berichtskomponente übergibt. Das Plug-in-Konzept von List & Label erlaubt es, nahezu beliebige eigene Objekte im Designer einzubinden. So sind auch komplexe Spezialfunktionen kein Problem – dazu aber später mehr.

Den Holometric-Anwendern steht der Designer zur Verfügung, damit sie Projektvorlagen selbst gestalten können. Um den Einstieg zu erleichtern, liegen der Software einige Projektvorlagen bei, die in der Regel die Anforderungen der Kunden erfüllen.

Diese Vorlagen sind dank der Flexibilität von List & Label so gestaltet, dass sie vom Kunden auch ohne besondere Kenntnisse an die eigenen Bedürfnisse anzupassen sind.

Schnelle Ergebnisse

Andreas Glaubitz, Anwendungsentwickler bei Holometric Technologies, kam bei der Evaluierung mit dem Einstieg in List & Label schnell zurecht. Dabei half auch die gute Dokumentation sowie die Unterstützung durch combit. Die Grundkonzepte von List & Label ließen sich schon

Auf einen Blick

Autor



Jochen Bartlau ist seit 1998 beim Konstanzer Softwareunternehmen combit beschäftigt. Als Projektleiter ist er zusammen mit seinem Team für die technische Weiterentwicklung der Berichtskomponente List & Label verantwortlich.

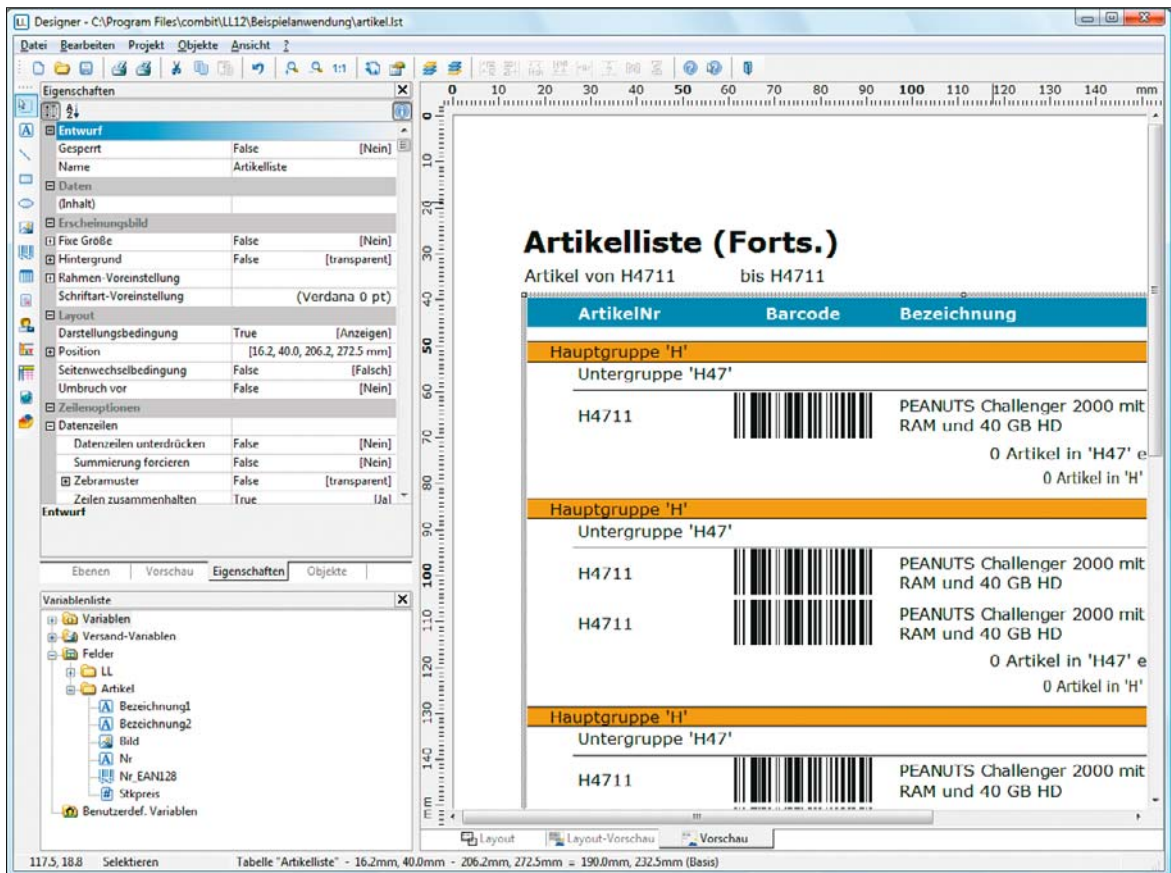
dotnetpro.code
A0706ListLabel

Sprachen C#

Technik Berichte

Voraussetzungen List & Label (combit)

Abbildung 1
Der
List & Label
Designer.



beim Testen mit der Trialversion umsetzen, darum konnte er List & Label bereits nach kurzer Zeit effektiv verwenden.

Nach dem erfolgreichen Test eines Prototyps und der Entscheidung für die Komponente wurden in zwei Workshops – einem für die Entwickler und einem für die Report-Designer – die Grundlagen für die Implementierung erarbeitet.

Die Entscheidung hat sich für Holo-metric Technologies bezahlt gemacht: Andreas Glaubitz schätzt, dass durch den Zukauf etwa 70 Prozent Aufwand und Zeit gespart werden konnten. Ohne List & Label könnte das Produkt zudem heute nicht die umfangreichen Export- und Gestaltungsmöglichkeiten bieten.

„Mir als Entwickler hat vor allem die einfache Integration von List & Label in bestehende Applikationen gefallen. Auch beim Erstellen von Projektvorlagen bietet List & Label viele Möglichkeiten“, so Glaubitz.

Wie es gefällt

Kein zugekauft Produkt kennt die Geschäftslogik einer Anwendung so gut wie sie selbst. Deshalb stößt die Berichts-ausgabe oft an Grenzen und es ist nötig, Werte dann doch mühsam selbst zu übergeben – oder, wenn das verwendete Tool keine ungebundenen Daten unterstützt, sogar mit berechneten Feldern in der Da-

tenbank zu arbeiten. Der List & Label Designer dagegen zeigt sich offen gegenüber Erweiterungen. Die Anwendung kann zum Beispiel eigene Funktionen an die Komponente übergeben. Sie stehen dann dem Anwender wie die schon vorhandenen Funktionen zur Verfügung und sind von diesen nicht zu unterscheiden. List & Label stellt dazu eine komplette Infrastruktur zur Verfügung, siehe Abbildung 3.

Die Anwendung muss lediglich die eigentliche Berechnung vornehmen und kann sogar die Vervollständigung der Parameter innerhalb des Designers steuern (siehe Listing 1).

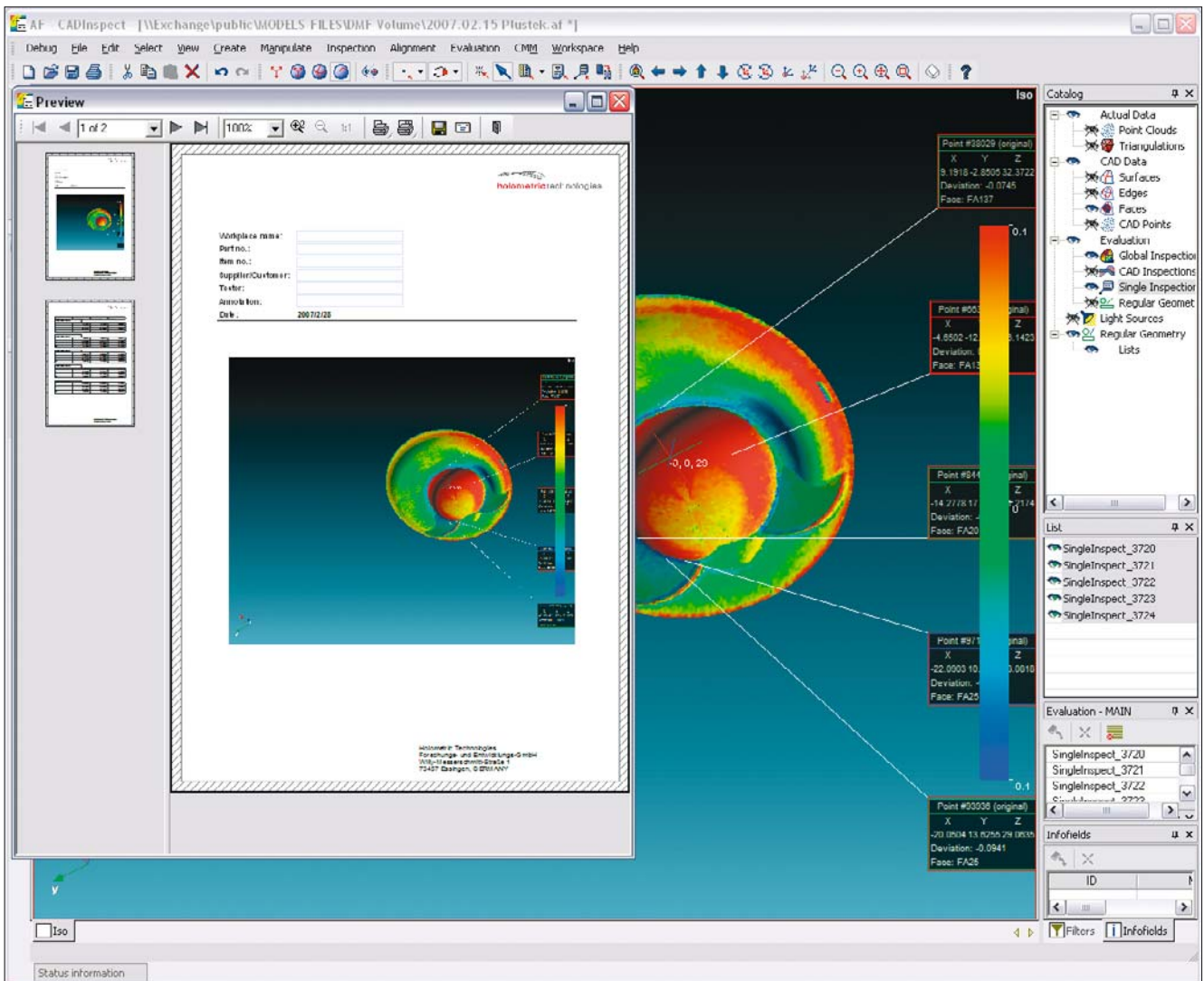
Daneben lassen sich im Designer vorhandene Funktionen auch ausblenden,

Listing 1

Eine einfache benutzerdefinierte Funktion.

```
private void designerFunktion1_EvaluateFunction(object sender, combit.ListLabel12.EvaluateFunctionEventArgs e) {
    e.ResultValue = (double.Parse(e.Parameter1.ToString()) + double.Parse(e.Parameter2.ToString()));
}

private void designerFunktion1_ParameterAutoComplete(object sender, combit.ListLabel12.ParameterAutoCompleteEventArgs e) {
    for(int i = 1; i<=20; i++)
        e.Values.Add(i.ToString());
}
```



wenn die Devise „Konzentration auf das Wesentliche“ lautet.

Das Sperren sensibler Objekte, das Ausblenden einzelner Menüpunkte und eine anwendergerechte Benennung der Datenfelder kann die Hürde für Layoutanpassungen auch für ungeübte Anwender verkleinern.

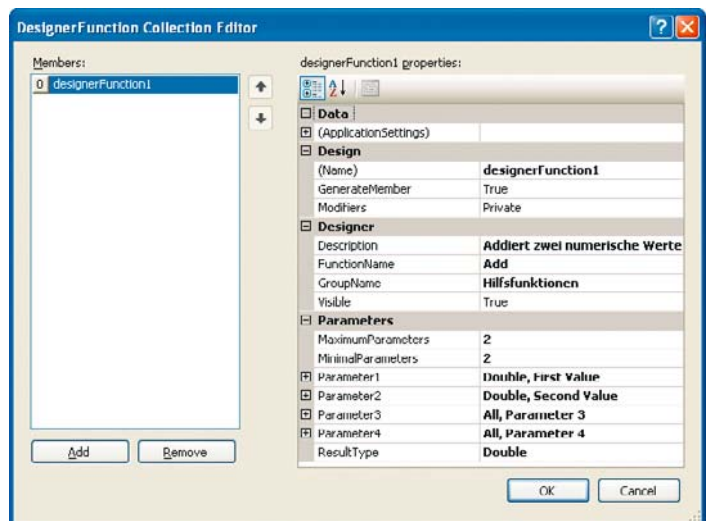
Auch eigene Objekte kann die Anwendung im Designer anmelden. Gerade wenn Visualisierungen sehr komplex werden oder Ausgaben aus Spezialanwendungen zu übergeben sind, ist eine solche Möglichkeit von großem Wert.

Fazit

Auf Ausgabemöglichkeiten kann keine Anwendung mehr verzichten. Durch ein Berichtstool wie List & Label lassen sich bestehende Applikationen schnell um viele entsprechende Funktionen ergänzen. Der Anwender muss dabei gar nicht

Abbildung 2
List & Label eingesetzt im AF Application Framework von Holometric Technologies.

Abbildung 3
Der Collection-Editor für Funktionserweiterungen.



merken, dass hier im Hintergrund eine „fertige“ Komponente werkelt – ihre Anpassbarkeit und Erweiterbarkeit erlaubt es, fast jede gewünschte Funktion nach-

zurüsten. So wird aus der Einheitslösung von der Stange schnell ein Werkzeug, das genau an die eigenen Bedürfnisse angepasst ist. |||||