

## Interaktive Diagramme mit Flash und PHP

# Hübsche Zahlen

Die Visualisierung von Kennzahlen stellt eine wesentliche Erleichterung bei der Entscheidungsfindung in der Steuerung von Unternehmen und ihren Prozessen dar. Beliebt sind hierfür z. B. Tortendiagramme oder Darstellungen von Säulen, die mit den verbreiteten Tabellenkalkulationen MS Excel oder Open Office einfach realisiert werden können. Doch wie publiziert man sich häufig ändernde Daten in kurzen Zeitintervallen und präsentiert sie in ansprechender Weise auch einer größeren Benutzergruppe?

von Ralf Hohoff

Open Flash Chart (OFC) ist eine Open-Source-Lösung, die zur Visualisierung das weit verbreitete Flash-Format nutzt und hiermit auch noch ein gewisses Maß an Dynamik und Interaktivität bietet. Es stehen verschiedene Kurven-, Säulen- und Tortendiagramme zur Auswahl, die die häufigsten Anwendungsszenarien abdecken. Irritierend wirkt bei den Beispielen auf der Projekthomepage die eigenwillige Farbwahl, der Charme der frühen Neunziger macht sich breit – vielleicht ein Ansporn, Ansprechenderes zu erstellen. Die verfügbaren Konfigurationsmöglichkeiten lassen hierfür auf jeden Fall wenige Wünsche offen. Wer Flash und ActionScript beherrscht, kann sogar die Flash-Quelldateien (\*.fla) modifizieren, denn auch diese stehen unter einer Open-Source-Lizenz zur Verfügung. Um die Chart Engine in eine Website einzubinden, zu konfigurieren und mit Daten zu speisen, sind einfache Programmierkenntnisse ausreichend. Im Zentrum steht die Flash-Anwendung, die ihre Vorgaben zur Visualisierung und die Daten selbst aus einer separaten Datei erhält. Diese separate Steuerdatei für Stil und Inhalt des Diagramms kann man von Hand erstellen – muss man aber nicht.

Die geforderte Dynamik lässt sich eleganter abbilden, indem man die Datei über die mitgelieferten Bibliotheken für PHP, Perl, Python, Java oder Ruby on Rails erstellt. Für die Steuerdatei wurde mit Version 2 von Open Flash Chart das Format JavaScript Object Notation (JSON) eingeführt. JSON ist kompakter und schlanker als XML, und obwohl der Name auf eine alleinige Verwendung in JavaScript hindeutet, ist JSON ein unabhängiges Format zum Datenaustausch zwischen Anwendungen. Ein weiterer Vorteil in der Entwicklung ist, dass mit JSON die Datei für Mensch und Maschine lesbar ist. Der verborgene Charme der Chart Engine liegt in diesem zweigeteilten Aufbau zwischen Darstellung und Daten. Bei OFC wird weder ein Flash-Film dynamisch auf dem Server generiert, noch aus ihm heraus eine direkte Datenbankabfrage ausgeführt. OFC stellt eine Flash-Anwendung bereit und überlässt dem Entwickler die freie Wahl, wie die Daten bereitgestellt werden. Das hat mehrere signifikante Vorteile:

- Je nach Implementierung wird das Shockwave Flash Object (SWF) mit einer Dateigröße von 270 KB nur einmalig vom Server heruntergeladen und vom Browser zwischengespeichert.

- Die Steuerdateien können in verschiedensten Programmiersprachen oder aus Drittsystemen wie Content-Management-Systemen heraus erstellt werden.
- Die Datenselektion und -aggregation auf dem Server steigert Performance und Sicherheit.
- Eine Zwischenspeicherung (Caching) der erstellten Dateien (Größe üblicherweise zwischen 1 bis 3 KB) auf dem Server ist einfach zu realisieren.
- Aktualisierung der Charts ohne Reload der Seite möglich (AJAJ).
- In einer Webseite können mehrere Charts mit nur einer Flash-Anwendung angezeigt werden.

## Geteilte Dynamik

Nach dem Download der aktuellen Version von [1] wird das Zip-Archiv in unserem Installationsverzeichnis *ofc2* entpackt. Direkt in diesem Verzeichnis befindet sich nun die Flash-Datei *open-flash-chart.swf*, mit der die verschiedenen Diagrammtypen visualisiert und animiert werden. In Unterverzeichnissen sind Beispiele für statische Steuerdateien (Data Files) und die oben genannten Bibliotheken für die gebräuchlichsten Programmiersprachen abgelegt.

Als übersichtliche Beispielaufgabe werden wir fiktive Umsatzzahlen im Vergleich zwischen Vorjahr und aktuellem Jahr als Balkendiagramme darstellen. Für den Anfang legen wir in dem neuen Unterordner */pm/* ein einfaches HTML-Dokument an und betten die Flash-Datei ein, wie es Listing 1 zeigt. Der teilweise „doppelte“ Quelltext mit *object*- und *embed*-Tags ist notwendig, da der Internet Explorer und die anderen Browser unterschiedlich angesprochen werden müssen. Belassen wir es für den Start dabei, später „rationalisieren“ wir Überflüssiges weg. Wenig überraschend zeigt ein erster schneller Seitenaufruf keinen Inhalt bis auf eine Fehlermeldung. Es fehlen noch alle Angaben zu dem gewünschten Diagrammtyp und die Daten selbst sind auch nicht da.

Diese Aufgabe übernimmt unser Skript *turnovers.php* (Listing 2), es startet mit dem Laden der entsprechenden PHP-Library. OFC ist in PHP objektorientiert umgesetzt und initialisiert mit *\$chart = new open\_flash\_chart();* ein neues Chart-Objekt. Direkt im Anschluss wird der Titel vergeben und ebenso wie der Wert für die Hintergrundfarbe dem Chart mitgeteilt.

In einem Chart können mehrere Diagramme enthalten sein, in Schritt zwei und drei initiiert *\$bar = new bar\_glass()* die Säulendiagramme. Jedes Diagramm erhält mit *\$bar->colour* eine spezifische Balkenfarbe und die Beschriftung für die Legende übernimmt *\$bar->key*. Zur besseren Übersicht sind die Umsatzdaten unseres Beispiels direkt in Arrays hinterlegt, üblicherweise werden die Werte aus Datenbanken oder anderen Quellen gefüllt. Mit dem Befehl *set\_values* werden die Umsatzjahreswerte dem jeweiligen Säulendiagramm übergeben.

Mit der Funktion *set\_tooltip* wird definiert, welche Informationen angezeigt werden, wenn der Mauszeiger über einem Balken steht. In dem Anzeigetext können „magische“ Variablen eingefügt werden, die die Flash-Anwendung dynamisch durch die entsprechenden Werte ersetzt. Zur Verfügung stehen der Wert selbst (*#val#*), Informationen zu X- und Y-Koordinaten (*#x\_label#*, *#x#*, *#y#*) und Datumsangaben (*#date: d.m.Y#* mit der PHP-Datumsformatierung [2]). Neben

dem Text und den Variablen kann auch das Aussehen des Hinweistexts angepasst werden, z. B. Rahmenstärke und Hintergrundfarbe. Über *\$chart->add\_element(\$bar1)* bzw. *\$bar2* werden die vordefinierten Diagramme dem Chart hinzugefügt. Danach wird noch die Beschriftung der X-Achse und deren Darstellung definiert. Danach erfolgt die Ausgabe – und dieses Ergebnis kann sich wirklich sehen (lesen) lassen, denn es wird für uns mit

Zeilenumbrüchen und Einrückungen attraktiver gemacht (Listing 3).

An dieser Stelle kann evtl. ein Caching-Mechanismus zur Performancesteigerung angesetzt werden. Bei stark frequentierten Seiten mit entsprechend vielen Anfragen auf die Charts wird die JSON-Ausgabe temporär in einer Datei gespeichert. So werden vor allem die „langsamen“ Datenbankzugriffe stark minimiert – der DB-Administrator wird es uns danken.

### Listing 1

#### chart.html

```
<html>
<head>
<title>Open Flash Chart, Example 1</title>
</head>
<body style="font-family:verdana, sans-serif;">
<p>PHP Magazin</p>
<object classid="clsid:d27c6b6e-ae6d-
11cf-96b8-444553540000"
codebase="http://fpdownload.
macromedia.com/pub/shockwave/cabs/
flash/swflash.cab#version=8,0,0,0" width="600"
height="400" id="graph-2" align="middle">
<param name="allowScriptAccess"
value="sameDomain" />
</object>
</body>
```

```
<param name="movie" value="../../../open-flash-chart.
swf" />
<param name="quality" value="high" />
<embed src="../../../open-flash-chart.swf"
quality="high"
bgcolor="#FFFFFF"
width="600"
height="400"
name="open-flash-chart"
align="middle"
allowScriptAccess="sameDomain"
type="application/x-shockwave-flash"
pluginspage="http://www.macromedia.com/go/
getflashplayer" />
```

### Listing 2

#### turnovers.php

```
<?php
include '../php-ofc-library/open-flash-chart.php';
// Begin Chart
$chart = new open_flash_chart(); // new chart
$title = new title('2009 vs. 2010'); // define title
$chart->set_title($title); // set the title
$chart->set_bg_colour('#FFFFFF'); // set the
// background colour
// First bar chart
$bar1 = new bar_glass();
$bar1->colour('#B8BCC2');
$bar1->key('last year', 14); // legend text with height in
// pixels
$data1 = array(6,5,7,1,1,3,7,0,5,5,6,4,5); // read from
database
$bar1->set_values($data1);
$bar1->set_tooltip('turnover 2010: #val# Mio. EURO');
#$bar1->set_on_click('bar_2'); # Tooltip
$chart->add_element($bar1);
// Second bar chart
$bar2 = new bar_glass();
$bar2->colour('#4583DD');
$bar2->key('current year', 14); // legend text with
// height in pixels
$data2 = array(3,3,9,2,8,3,2,2,6,4,2,2); // read from
// database
$bar2->set_values($data2);
$bar2->set_tooltip('turnover 2010: #val# Mio. EURO');
#$bar2->set_on_click('bar_2'); # Tooltip
$chart->add_element($bar2);
// x-axis
$months = array('Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul', 'Aug',
'Sep', 'Oct', 'Nov', 'Dec');
$x_labels = new x_axis_labels();
$x_labels->set_steps(1); // display label on every step
$x_labels->set_vertical();
$x_labels->set_colour('#A2ACBA');
$x_labels->set_labels($months);
// x-axis
$x = new x_axis();
$x->set_colour('#A2ACBA');
$x->set_grid_colour('#D7E4A3');
$x->set_steps(1);
// add X Axis Labels to X Axis
$x->set_labels($x_labels);
// add x-axis to chart
$chart->set_x_axis($x);
// Output
echo $chart->toPrettyString(); // human readable
instead of $chart->toString();
?>
```

Um dem Flash-Film mitzuteilen, dass für uns die Quelle *turnovers.php* das Maß aller Dinge ist, übergeben wir diese Information als *GET*-Parameter *ofc* im Seitenaufruf, in unserem Beispiel *http://<servername>/ofc2/pm/chart1.html?ofc=turnovers.php*.

Der Browser lädt mit der Anfrage die HTML-Seite vom Server, interpretiert die Flash Tags und lädt die Flash-Anwendung *open-flash-chart.swf* nach. Open Flash Chart startet und ermittelt aus dem Seitenaufruf den Namen und Pfad der Steuerdatei *turnovers.php* und fragt diese an. OFC liest die Beschreibungsdatei ein und zeigt das Chart an, das Ergebnis zeigt Abbildung 1. Hinweis: Die Pfadangabe zur JSON-Datei ist immer absolut bzw. relativ zu dem HTML-Dokument und nicht zur SWF-Datei.

### AJAJ anstatt AJAX

Für einige der oben beschriebenen Funktionalitäten (Reload, mehrere Charts) wird der ursprüngliche HTML-Quellcode durch Funktionsaufrufe in JavaScript (JS) ersetzt. Da als Datenformat JSON und nicht XML genutzt wird, ist die Konzeptbezeichnung analog zu AJAX mit AJAJ (Asynchronous JavaScript and JSON) treffend. Die Bibliothek *js/swfobject.js* ist im Installationsverzeichnis von OFC bereits enthalten. Dieses JS steigert die Übersichtlichkeit enorm, denn es geht nun auf die Befindlichkeiten der unterschiedlichen Browser ein. Mit dem Funktionsaufruf *swfobject.embedSWF* wird an der Position des *<div>*-Elements der browserspezifische Code zum Aufruf des OFC (Listing 4) geschrieben.

Die Angabe der Steuerdatei ("*data-file*": "*turnovers.php*") ist optionaler Teil

des Funktionsaufrufs, alternativ ist außerdem auch der *GET*-Parameter möglich. Mit der Ansteuerung über JavaScript ergeben sich weitreichende neue Möglichkeiten. JS kann unter anderem die Aktualisierung des Charts übernehmen, ohne dass die gesamte HTML-Seite mit dem Flash-Objekt neu geladen werden muss. Dafür brauchen wir nur zwei neue Funktionen, *findSWF()* und *reload()*, die wir mit *<script type="text/JavaScript" src=".reload.js"></script>* einbinden (Listing 5).

Wird *reload()* über einen Link aufgerufen, ermittelt zuerst *findSWF()* abhängig vom Browser das Flash-Objekt und weist es danach an, die Steuerdatei neu zu laden. Der Einsatz der Reload-Funktionalität eignet sich hervorragend bei Daten, die sich schnell ändern. Dashboards mit stundenaktuellen Informationen (Stichwort „Intraday-Steuerung“) leben z. B. von kurzen Aggregationsintervallen. Neben der manuellen Aktualisierung durch einen Link kann auch ein automatisches Neuladen in definierten Zyklen stattfinden – hierbei bitte die Performance des Systems beobachten und gegebenenfalls Caching einsetzen.

Da die Steuerdatei sowohl Diagrammtyp als auch die Daten enthält, kann mit

#### Listing 3

```
turnovers.php.json
{
  "elements": [
    {
      "type": "bar_glass",
      "colour": "#B8BCC2",
      "text": "last year",
      "font-size": 14,
      "values": [6, 5, 7, 1, 1, 3, 7, 0.5, 5, 6, 4, 5],
      "tip": "turnover 2008: #val# Mio. EURO",
      "on-click": "bar_1"
    },
    {
      "type": "bar_glass",
      "colour": "#4583DD",
      "text": "current year",
      "font-size": 14,
      "values": [3, 3, 9, 2, 8, 3, 2, 2, 6, 4, 2, 2],
      "tip": "turnover 2010: #val# Mio. EURO",
      "on-click": "bar_2"
    }
  ],
  "title": {
    "text": "2009 vs. 2010"
  },
  "bg_colour": "#FFFFFF",
  "x_axis": {
    "colour": "#A2ACBA",
    "grid-colour": "#D7E4A3",
    "steps": 1,
    "labels": {
      "steps": 1,
      "rotate": 270,
      "colour": "#A2ACBA",
      "labels": ["Jan", "Feb", "Mar", "Apr", "May", "Jun", "Jul",
        "Aug", "Sep", "Oct", "Nov", "Dec"]
    }
  }
}
```

#### Listing 4

```
chart2.html
<html>
<head>
<title>Open Flash Chart, Example 2</title>
<script type="text/javascript" src=".../js/swfobject.js">
</script>
<script type="text/javascript">
swfobject.embedSWF(
  ".../open-flash-chart.swf", "my_chart", "600", "400",
  "9.0.0", "expressInstall.swf",
  {"data-file": "turnovers.php"}
);
</script>
<script type="text/javascript" src=".../reload.js">
</script>
<script type="text/javascript" src=".../tooltip.js">
</script>
</head>
<body style="font-family: verdana, sans-serif;">
<p>PHP Magazin</p>
<div id="my_chart"></div>
<a href="javascript:reload()">Reload data only</a>
</body>
</html>
```

#### Listing 5

```
reload.js
function findSWF(movieName) {
  if (navigator.appName.indexOf("Microsoft") != -1) {
    return window[movieName];
  } else {
    return document[movieName];
  }
}
function reload(){
  tmp = findSWF("my_chart");
  x = tmp.reload("turnovers.php");
}
```

#### Listing 6

```
tooltip.js
function bar_1( month ){
  showinfo(1, month);
}
function bar_2( month ){
  showinfo(2, month);
}
function showinfo( bar, month ){
  alert('bar' + bar + ', month' + month);
}
```



der Funktionalität des Reloads nicht nur eine Aktualisierung desselben Charts durchgeführt, sondern auch ein vollständiges neues Chart geladen werden. Mit einer Erweiterung des JavaScripts und zusätzlichen Links zum Aufrufen von verschiedenen Datenquellen können so an einer Position in der Webseite unterschiedliche Charts dargestellt werden. Vorteil ist, dass weder die HTML-Seite selbst noch die „große“ Flash-Datei bei einem Wechsel der Charts neu geladen werden müssen.

Aber auch die gleichzeitige Darstellung von mehreren Charts auf einer HTML-Seite ist mit den bereitgestellten JavaScript-Funktionalitäten nun eine einfach zu realisierende Aufgabe. Insbesondere bei der Einbindung der OFC Chart Engine in vorhandene Systeme wie CMS ist das der bevorzugte Weg, denn für jedes Chart wird nur der JS-Funktionsaufruf `swfobject.embedSWF()` und ein korrespondierendes `<div>`-Element benötigt.

### Es geht auch anders

Die Interaktion beschränkt sich aber nicht nur auf die Richtung vom JavaScript zum Chart, auch aus OFC heraus können umgekehrt JS-Funktionen oder externe Webseiten aufgerufen werden. Ergänzen wir das PHP-Skript um `$bar1->set_on_click()` bzw. `$bar2->set_on_click()` (jeweils vor `$chart->add_element()`), weisen wir Flash damit an, bei Klick auf die Diagrammbalken eine Aktion auszuführen. Beginnt die Zeichenkette in `set_on_click` mit `http` oder `https` wird OFC den URL öffnen, ansonsten wird es versuchen, die entsprechende JavaScript-Funktion aufzurufen. Über `<script type="text/JavaScript" src="/tooltip.js"></script>` wird auch diese Funktionalität in unsere HTML-Seite eingebunden (Listing 6). In unserem Beispiel werden aus Flash heraus die JS-Funktionen `bar_1` bzw. `bar_2` aufgerufen, um einen statischen Parameter je Diagramm ergänzt und über `showinfo` als MessageBox dem Benutzer angezeigt.

Da bei den JavaScript-Aufrufen Chart-Parameter mitgegeben werden, können diese auch genutzt werden, um weitere Funktionalitäten abzubilden. Vorstellbar ist das dynamische Einblen-

Abb. 1: Anzeige des fertigen Charts



den weiterführender Charts nach Zeiträumen oder beliebigen anderen Dimensionen (Stichwort „Drilldown“).

Dieser Einstieg zeigt einen Überblick der weitreichenden Möglichkeiten von Open Flash Chart. Viele kleine Details finden sich noch in den umfangreichen Tutorials. Das Spektrum der möglichen Diagramme (Kuchen, Linien, Radar) ist vielfältig und die Darstellung kann umfassend nach den eigenen Vorstellungen angepasst werden. Hier noch ein kleiner Tipp: Nutzt man bei den kleinen Herausforderungen im Programmieralltag für die Recherche eine Suchmaschine, kann man schnell auch auf die Dokumentation der OFC-Vorgängerversion gelangen (die URLs unterscheiden sich nur marginal), die aber bei der zweiten Version wenig hilfreich ist.

### Fazit

Open Flash Chart ist eine sehr flexible Komponente für die Erstellung von dynamischen Flash-Diagrammen. Verschiedenste Darstellungsarten werden unterstützt, nur für Tachoanzeigen muss man – noch – auf andere Tools wie Open Flash Gauges [3] zurückgreifen. Die vielfältigen Konfigurationsmöglichkeiten lassen fast keine Wünsche offen, alle Daten und Parameter für Farben oder Beschriftungen werden durch die konsequente Objektorientierung in der PHP-Bibliothek strukturiert übergeben. Die Architektur mit einer „intelligenten“ Flash-Anwendung und der Steuerdatei im JSON-Format ist wohl konzipiert, denn durch die vielen mitgelieferten Softwarebibliotheken kann jede aktuelle Programmiersprache die Datei

im gewünschten Format erstellen, ohne sich dabei mit einem XML-Overhead abzumühen. Steigt die Anzahl der Seitenzugriffe, kann die Performance über ein Caching der Steuerdateien auf dem Server auf hohem Niveau gehalten werden. Die bidirektionale Interaktion zwischen Flash und JavaScript bietet viel Potenzial für die individuelle Gestaltung eigener Anwendungen. Das stellt auch den wesentlichen Unterschied zu Programmen wie pchart [4] dar, mit dem man „nur“ dynamisch Bilder erstellen kann. Der beachtliche Funktionsumfang in Kombination mit der großen Flexibilität bei der Integration macht Open Flash Chart wohl auch so beliebt. Für zahlreiche Systeme wie Drupal oder Ruby on Rails existieren bereits Erweiterungen.



**Ralf Hohoff** ist seit mehr als 10 Jahren in der Softwareentwicklung und dem Projektmanagement tätig. Als technischer Berater der buw consulting GmbH aus Osnabrück unterstützt er Kunden bei der Konzeption und Implementierung von IT-Systemen mit den Schwerpunkten CRM und Wissensmanagement.

### Links & Literatur

- [1] Aktuelle Version von Open Flash Chart: <http://teethgrinder.co.uk/open-flash-chart-2/downloads.php>
- [2] PHP-Datumsformatierung: <http://www.php.net/date>
- [3] Open Flash Gauges: <http://www.darkartistry.com/content/view/129/48/>
- [4] pchart: <http://pchart.sourceforge.net/index.php>