

ВИЗУАЛИЗАЦИЯ ГРАФИЧЕСКИХ ДАННЫХ В ПРИЛОЖЕНИИ ASP.NET MVC

Хасенов Е.А.¹, Санкибаев А.Т.²

¹Хасенов Есиль Адикович – кандидат физико-математических наук, доцент;

²Санкибаев Арман Темирханович – магистрант,
кафедра информатики и информационной безопасности,
Евразийский национальный университет им. Л.Н. Гумилева, г. Астана, Республика Казахстан

Аннотация: в статье рассматривается проблема визуализации больших объемов данных, которая заключается в создании модуля для информационно-аналитической системы, с помощью средств веб-разработки. Показан подход к решению задачи визуализации больших объемов данных, обеспечивающий анализ и правильную интерпретацию результатов вычислений, а также дальнейшую работу с вычислительной моделью.

Ключевые слова: визуализация, график, ASP .NET MVC, Highcharts, jQuery.

УДК 004.81: 004.89

Построение моделей – универсальный способ изучения окружающего мира, позволяющий обнаруживать зависимости, прогнозировать, разбивать на группы и решать множество других задач. Основная цель моделирования в том, что модель должна достаточно хорошо отображать функционирование моделируемой системы [1].

Чтобы получить новые знания об исследуемом объекте или явлении, не обязательно строить сложные модели. Часто достаточно посмотреть данные в нужном виде, чтобы сделать определенные выводы или выдвинуть предположение о характере зависимостей в системе, получить ответ на интересующий вопрос. Это помогает сделать визуализация [1].

Все большую популярность приобретает язык JavaScript как средство визуализации данных на веб-интерфейсе. На текущий момент существует много JavaScript - библиотек визуализации данных.

Одной из самых популярных библиотек для построения графиков с помощью JavaScript является Highcharts. В Highcharts имеется много удобных функций и дополнений. Поддерживается во всех основных браузерах: Mozilla Firefox, Google Chrome, Safari, в Internet Explorer. С графиками можно взаимодействовать: менять масштаб, отключать ненужные графики или подсвечивать необходимые, добавлять новые точки, использовать обработчики нажатий кнопки мыши, надписи и т.п.; входные данные можно считывать из локального файла, использовать AJAX, обращаться к БД, получать в формате JSON. Также путем манипуляций полученные графики можно сохранять в PNG картинки. Очень удобно построен экспорт из браузера – библиотека реализует экспорт в формате JPEG, PDF, SVG [2].

Рассмотрим конкретный пример визуализации данных с помощью библиотеки Highcharts на примере отображения динамики изменения количества фактов пресечения попытки провоза наркотических средств через государственную границу. Реализация будет осуществляться с фреймворком ASP.NET MVC на платформе Microsoft Visual Studio 2012. Отобразим статистику по количеству фактов по двенадцати месяцам.

На первом этапе создадим html-страницу с div-элементом. Затем создадим, добавим ссылки на JavaScript-библиотеки Highcharts, jQuery [3].

```
<script src="@Url.Content("~/Scripts/highcharts/highcharts.js")" type="text/javascript"></script>
```

```
<script src="@Url.Content("~/Scripts/jquery-ui/jquery-ui.min.js")" type="text/javascript"></script>
```

```
<div id="calendar-chart">
```

```
<div id="datepicker" style="font-size:13px;"></div>
```

```
<div id="bar_chart" style="margin: 0 auto;"></div>
```

```
</div>
```

Создадим JavaScript-функцию, которая будет загружать график на html-страницу, через метод подгрузим данные динамически в график.

```
$(function () {  
    $('#bar_chart').highcharts({  
        chart: {  
            type: 'line'  
        },  
        title: {  
            text: 'Всего с начала года'  
        }  
    });  
});
```

```

    },
    subtitle: {
      text: 'Источник: База данных'
    },
    xAxis: {
      type: 'category',
      labels: {
        rotation: -45,
        style: {
          fontSize: '13px',
          fontFamily: 'Verdana, sans-serif'
        }
      }
    },
    plotOptions: {
      series: {
        color: '#FF0000'
      }
    },
    yAxis: {
      min: 0,
      title: {
        text: 'Количество'
      }
    },
    legend: {
      enabled: false
    },
    tooltip: {
      pointFormat: 'Всего: <b>{point.y:.1f}</b> единиц(а)</b>',
    },
    series: [{
      name: 'Population',
      data: [],
      dataLabels: {
        enabled: true,
        rotation: -90,
        color: '#FFFFFF',
        align: 'right',
        x: 4,
        y: 10,
        style: {
          fontSize: '13px',
          fontFamily: 'Verdana, sans-serif',
          textShadow: '0 0 3px black'
        }
      }
    }
  ]
});
});
$.ajax({
  url: '@Url.Action("getDataByQual")',
  data: "qual_id=" + qual_id + "&startdate="+start + "&enddate="+end,
  dataType: 'JSON',
  success: function (data) {
    var chart = $('#bar_chart').highcharts();
    var new_data = JSON.parse(data);
    chart.series[0].remove();
    chart.addSeries({data: new_data});
  }
});

```

```
});
```

Как видим из кода, данные загружаются динамически из метода контроллера `getDataByQual`, реализованного на языке программирования `C#`. Данный метод создает объект типа `chartModel`, который, обернутый в `JSON`-формат, передается в `JavaScript`-функцию. То есть, вся логика `MVC` обрабатывается в контроллере и интерфейс получает только готовый численный результат.

```
public JsonResult getDataByQual(int qual_l2_id)
{
    //передается в виде json для графика
    chartModel data = new chartModel();
    //получаем дату начала года
    var start = DateTime.Now;
    //Смещаем месяц плюс один для получения начала предыдущего года
    start = start.AddMonths(-start.Month+1);
    start = start.AddDays(-start.Day);

    // формируем запрос
    var facts = db.M_FACT.Where(i.SYSTIME >= start && i.FACTTYPE_ID == 10);
    //месяца
    var monthes = new string[] { "янв", "фев", "мар", "апр", "май", "июнь", "июль", "авг", "сен",
"окт", "ноя", "дек" };
    //считаем данные по регионам
    var counts = facts
        .GroupBy(l => new { Month = l.SYSTIME.Month })
        .Select(g => new { Month = g.Key.Month, Count = g.Select(l => l.ID).Count() })
        .ToDictionary(o => o.Month, o => o.Count);
    // наполняем данными по именам регионов
    data.data = new List<IList<object>>>();
    for (var i = 0; i < monthes.Length; i++)
    {
        if (DateTime.Now.Month < i+1) break;

        if (counts.ContainsKey(i)) data.data.Add(new List<object>(new object[] { monthes[i], counts[i]
    }));
        else data.data.Add(new List<object>(new object[] { monthes[i], 0 }));
    }
    //цвет в зависимости от уровня
    color="#0000FF";
    data.color = color;
    return Json(data, JsonRequestBehavior.AllowGet);
}
```

Таким образом, в результате формируем массив месяцев с численными показателями (Таблица 1).

Таблица 1. Численные показатели

Январь	32
Февраль	12
Март	7
Апрель	25
май	16
июнь	25
июль	15
август	39
сентябрь	39
октябрь	4
ноябрь	5
декабрь	6

В итоге получим график с отображением динамики фактов по республике на рис. 1. Таким образом, с помощью библиотеки Highcharts мы получили наглядную диаграмму с численными параметрами, которая показывает изменение ситуации по времени.

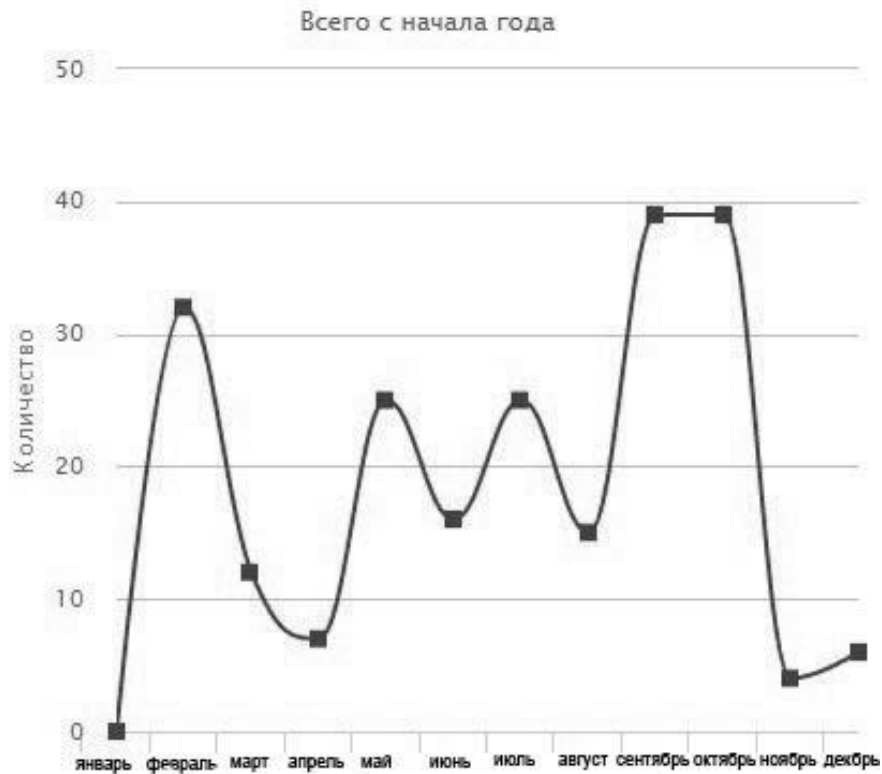


Рис. 1. График с отображением динамики количества фактов

Список литературы

1. Паклин Н.Б., Орешков В.И. Бизнес-аналитика от данных к знаниям. Питер, 2013. 20-27 с.
2. Чекко Р. Графика на JavaScript: O'Reilly, 2014. № 9. С. 15–16.
3. Highcharts Demos / Spline with symbols. [Электронный ресурс]. Режим доступа: <http://www.highcharts.com/demo/spline-symbols/> (дата обращения: 02.04.2017).