

# 基于 HTML5+nodeJS+MongoDB 构建在线图像编辑器系统

柯肇丰, 曾霞霞

(闽江学院计算机科学系 福建 福州 350108)

**【摘要】**针对在线图像编辑器系统的功能需求,解释了系统开发使用的 HTML5 和 JavaScript 技术、以及 nodeJS 构建 Web 服务的特点与 MongoDB 数据存储的优势。通过介绍系统界面的设计、图层构建方法、Web 后端服务实现和 MongoDB 数据结构,展示了系统的友好用户界面,论证了技术方案的可行性。

**【关键字】**图像编辑器;HTML5;JavaScript;nodeJS;MongoDB

## 0 引言

提到图像编辑器大家并不陌生,市场上流行软件有美图秀秀、QQ 影像等。所谓的在线,主要指的是 Web 版的,网页版的美图秀秀就是个典型的在线图像编辑软件。本文要开发的在线图像编辑器系统主要应用电子商务平台,适用于客户在线订制产品的需求。举例说明下,比如手机壳生产商提供一个平台,客户可以在线选择自己要订制的产品的规格、型号,以及可以使用这个平台在线设计一下产品的外观并展示效果。这个平台指的就是在线图像编辑器。

## 1 在线图像编辑器介绍

### 1.1 系统功能设计

图像编辑器系统主要实现用户在线选择预设计的模型,然后在模型图上通过选择背景、添加饰品、编辑文字等处理,最终在结果展示窗口上显示编辑效果并保存作品的过程。所以我们需要设计的图像编辑器,它并不需要像 Photoshop 那样在空白画板上进行从无到有的图画设计,而是在某种模型如手机壳、相框等图板上添加各种素材图层来组合完成最终作品的图案编辑的过程。编辑器软件的功能需求分用户中心和模型设计两大模块,用户中心包含用户信息管理、素材上传、作品管理三个子模块;模型设计是系统的核心功能,包含模型选择、背景图层、文字图层、饰品图层、表情图层、自选素材等菜单栏目,以及预览和保存功能。

### 1.2 开发技术的选择

该系统是基于 B/S 的架构的 Web 版的应用,使用 JavaScript 作为主要开发语言,使用第三方 RaphaelJS 库提供的一些工具来实现 SVG 图形的绘制。结合 HTML5 的新技术构思和建设网页,nodeJS 技术的应用使得 JavaScript 可以构建高性能的后端服务,数据信息的管理更是大胆的使用了非关系型数据库 MongoDB,面向集合的数据存储能够更加灵活、高效和可扩展的处理该软件需要处理的复杂的素材信息。

## 2 相关技术概述

### 2.1 HTML5 与 JavaScript 简介

随着 WEB 技术的发展,动态网站和在线应用程序开始被广泛应用,传统的在线图像编辑器多数使用的是内嵌 Adobe Flash 和第三方插件控制的方法,而 HTML5 的出现,使得在线应用程序的开发更加的灵活。HTML5 出现之前,浏览器本身是

不支持图形图像处理,只能以嵌入编辑好的图像的形式存在。HTML5 的 Canvas 和 SVG 使浏览器能够直接创建图形并处理图像。SVG 是一种使用 XML 描述 2D 图形的语言<sup>[1]</sup>,Canvas 可以看做一个 2D 的图形环境<sup>[2]</sup>,通过使用 JavaScript API 来操作 SVG 和 Canvas 元素,可以创建响应用户交互的图形和动画。本文所介绍的编辑器系统正是基于 SVG 图形和 Canvas 元素的应用而设计开发的。

JavaScript 是一种直译式脚本语言,其源代码在发往客户端运行之前不需经过编译,而是将文本格式的字符代码发送给浏览器由浏览器解释运行<sup>[3]</sup>。它已经被广泛用于 Web 应用开发,常用来为网页添加各式各样的动态功能,为用户提供更流畅美观的浏览效果。随着 HTML5 的到来,JavaScript 的生命力更加顽强,V8 引擎和 nodeJS 框架的发展,以及事件驱动及异步 IO 等特性,JavaScript 逐渐被用来编写服务器端程序。

### 2.2 nodeJS 技术的特点

在 nodeJS 是近几年才兴起的新技术,使用的是 JavaScript 语法,用于高性能后端服务的开发。nodeJS 的开发简单,只需几行代码就能写出一个简单的服务器程序<sup>[4]</sup>,同时由于它的非阻塞特性以及异步的事件驱动模型,在开发高并发网络服务时,性能非常突出<sup>[5]</sup>。nodeJS 的运行不需要 WebServer,程序执行的开销也少,因此 nodeJS 系统占用很少的系统资源,控制了运行的成本。nodeJS 的解释引擎基于 V8,V8 也是 google Chrome 的 JS 引擎,执行效率非常高<sup>[6]</sup>。正是缘于 nodeJS 的这些优点,选择了它作为图像编辑器软件的后端服务开发技术。

### 2.3 MongoDB 存储非结构数据

数据库技术发展至今产生了 SQL 数据库和 NoSQL(对所有非关系数据库的统称)数据库两大类型<sup>[7]</sup>。非关系型数据库不支持关系模型,不支持连接操作,易使用和部署,在实现海量数据分布式存储和快速方法技术上取得了一定的成果。MongoDB 是一个基于分布式文件存储的 NoSQL 数据库<sup>[8]</sup>,由 C++编写的,存储数据方便、性能高。这为本文研究来源于不同用户、不同类型、构建不同数据集的网络素材信息的管理提供了技术支撑。

## 3 在线图像编辑器界面设计

### 3.1 编辑器的工作主界面

编辑器的主界面主要由素材栏、画板编辑区域和预览效果区域组成,如图 1 所示。其中素材栏主要以可展开式菜单结构

分类展示了可用于编辑图案的各种素材,如文字素材、饰品素材、表情素材等。

主界面的结构是按模块进行布局的,设计了结构 XML 文档,把页面分 header、leftNav 和 mainContent 三区块。每个区块分别绑定不同的数据源网页,易于页面的修改和内容的更新,同时把页面设计与处理代码分离,提高开发效率。结构 XML 文档的代码设计如下所示。

```
<header>{{#include 'template/nav.html', '$This'}}</header>
<content>
<leftNav > {{#include 'template/material.html', '$DATA.explorer.0.
childs.0'}}</leftNav>
<mainContent >{{#include 'template/plane.html', '$This'}} </mainCon-
tent>
</content>
```



图1 工作主界面图

### 3.2 表情图层实例分析



图2 表情图层编辑状态图

各种图像、文字素材被当作图层来处理,那么图层的绘制是该项目开发的技术重点。绘图核心主要由 sketchpadManage 类为主要入口,进行管理多个 sketchpad 画布类实例,每个 sketchpad 画布实例代表着一块矩形绘图区域。每个 sketchpad 画布实例还有一个 LayerManage 来作为 Layer 图层实例的管理器,来管理画布上多个图层与图层的创建和删除。

Layer 图层的效果如图 2 所示,展示了表情图层的部分属性编辑,图层的渲染需要设定相关属性值来定义图层显示的名称、顺序、大小、位置等静态信息,以及包含图层中的图像的源文件地址或者是文字图层的字体、字号等相关属性。

### 4 关键技术分析与实现

#### 4.1 在线图层编辑功能设计

图像编辑器软件的核心功能就是在一个选择好的模板图层面上进行图案与文字的编辑。底层模板、图案、文字等都是图层,这些图层在 WEB 上的绘制主要通过绘制 SVG 图形来实现的。第三方 RaphaelJS 库提供的一些很好的方法来操作 SVG 图形,避免直接使用 Javascript 编程出现很多繁琐的代码。

使用 RaphaelJS 生成 Raphael 类对象,参数包含图层的 ID、宽、高,同时创建一个 newSketchpad 画布对象即一块矩形绘图区域。可以用这句代码来实现:var newSketchpad = Object.create(Raphael (newSketchpad\_id, width, height), SketchpadPro prototype)。可以使用 Raphael 的方法编辑画布区域的各种效果颜色、旋转、缩放等,编辑的效果以 SVG 格式存在客户端,还需将它转化为图像并被预览和存储。

SVG 图层的组合在 WEB 上的预览这里借助了 HTML5 的 Canvas。SVG 到 Canvas 需要使用一个 google 开发的插件 canvg,该插件的 canvg (canvasId,svgHtml) 方法可以将 SVG 转成 canvas。再调用 canvas.toDataURL()将 Canvas 转成了图片,并返回图片的所有内容数据,以.png 或.jpg 格式保存结果图像。canvg 使用方法给出一个实例:

```
canvg(canvas, $dataNode.html(), { // canvg 的第一个参数是 html5 的
Canvas 对象
```

```
ignoreMouse: true, // $dataNode.html()是 svg 数据
renderCallback: function() { // 第三个参数是 canvg 的渲染配置。
canvas.renderCallback();//执行渲染回调,canvas 导出 Image
}, ignoreAnimation: true})).
```

这里需要强调一下不是所有的浏览器都支持这种方法的使用,Internet Explorer 9、Firefox、Opera、Chrome 以及 Safari 支持 canvas 的使用。

#### 4.2 nodeJS 技术构建 Web 后端处理

nodeJS 的出现使得 Javascript 的生命力更大顽强,可以从 Web 前端开发延伸到后端,由于 nodeJS 的并发性能很好,所以其在 Web 开发中的运用越来越热,成为改变 Web 开发格局的新力量。nodeJS 的开发相对简单,只需要几行代码就能写出一个服务器程序。

下面通过一段程序来解释 nodeJS 如何实现前后端数据的投递。前端欲向后台 MongoDB 数据库请求新增一条用户信息,需要请求打开数据库集合 editor-user,调用 ./mongodb /core/common.js 文件定义的 common.update 方法,把从前端投递的用户信息更新到数据库中。代码设计如下:

```
require ("./core/config").dbname = " editor-user"; //需要访问的集合
editor-user
var ObjectId = require('mongodb').ObjectId;
var common = require("./core/common");
module.exports = function(userId, key, obj) { //对外接口的设计
var self = this;
var setter = {$push: {}}; //执行 mongodb 的指令,$push 意味着插入
setter.$push[key] = obj;
common.update({ //使用 common.js 的 update 方法更新数据库集合
_id: new ObjectId(userId)
}, setter, function(closer, lineNumber) {
self.response.send(200, JSON.stringify(obj)); //把前端 JSON 数据投递
到数据库
});}.
```

#### 4.3 MongoDB 文档模型数据结构设计

Mongodb 数据库属于文档模型的 NoSQL 数据库,文档是多个键值对的有序的组合,允许创建非结构化、任意类型的键值对,键不用事先定义、也不会固定不变。图像编辑器软件设计了用户 editor-user、产品 user\_product、素材图层 img-layer、文字图层 font-layer、自组素材 other-layer 等数据库集合。集合由文档组成,BSON 格式存储的 editor-user 集合的文档格式如下:

```
editor-user{
  "_id": "U_500zn6g",
  "name": "g4",
  "email": "g4@qq.com",
  "password": "2a78add5a131b6e60523",
  "create_time": 1430927174681.0}
```

集合的键值类型也很丰富有 String、Array、Document、Int32 等,如图 3 展示了 user\_product 集合部分键值的设置情况,正是利用了 Mongodb 的文档中可以包含文档类型的键值,可以很方便的存储一个编辑好的成品图像与组合设计需要的多个素材文档,也便于组合设计的还原与修改。



Field	Type
_id	String
name	String
image	Array
type	String
create_time	Int32
update_time	Int32
delete_time	Int32
status	Int32
comment	Int32
reply	Int32

图 3 user\_product 集合部分键值及类型

## 5 总结

本文设计的在线图像编辑器,采用 Html5+JavaScript 作为

前端页面开发的方法,使用了 JavaScript API 处理复杂的图层编辑、叠加和格式转换;后端使用了 nodeJS+MongoDB 数据库,实现了高效、灵活和方便的数据处理。该软件完成了预期的功能设计,系统稳定、可扩展,达到实时的要求。但由于 JavaScript 解释引擎的约束和 Html5 对浏览器的要求,系统不支持所有浏览器,应用范围有限;而且 nodeJS 属于新兴技术,大公司对它的支持和应用不多,其可靠性还有待论证。

## 参考文献:

- [1]郑美容. 基于 HTML5 的 Web 应用研究[J]. 工业控制计算机, 2014(12):64-67.
- [2]宋茂海,孔玉. 基于 JQuery 网页自动阅卷方案的设计与应用[J]. 医疗卫生装备,2014(06):69-71.
- [3]赵昆. 改变 web 开发格局的新技术 node.js[J]. 程序员,2011(7):124-126.
- [4]李臣龙,戴汶倬. MongoDB 和 Node.js 的邮件收发系统[J]. 电脑编程技巧与维护,2013(07):64-67.
- [5]张尧,甘泉等. 基于 MongoDB 的地理信息共享数据存储模型研究[J]. 测试,2014(08):147-150.
- [6]陈涛. MVVM 设计模式及其应用研究[J]. 计算机与数学工程, 2014(10):1982-1984.

## 作者简介:

曾霞霞(1981-),女,福建宁德人,闽江学院计算机科学系讲师,硕士。主要研究方向 WEB 数据挖掘。柯肇丰(1992-),男,福建漳州人,闽江学院计算机科学系,本科学士。主要研究方向 Web 应用开发。

(上接第 75 页)

课程设计采用 2~3 学生一组,利用 Matlab、OpenCV 和 VC++ 协同完成一个小型项目,完成项目后学生先分组讲解课程设计的成果,再学生自评和小组互评,教师结合项目设计报告进行排名打分。教师指导排名靠前的项目,经完善后去申报大学生研究性学习和创新性实验计划项目,或参加程序设计竞赛应用开发类作品大赛,这极大地促进了学生自主学习图像处理的动力,提高了合作和沟通水平<sup>[8]</sup>。

学生在完成小型项目开发和实现时,首先通过查阅资料选择该项目各处理步骤所用算法,再分析算法过程,最后编程实现。在此过程中,学生加深了对算法原理的理解,提高了程序设计能力,体会到利用理论知识解决实际问题的乐趣。

## 6 结语

数字图像处理是信息、电子和计算机等专业的一门极其重要的课程,既有抽象的理论推导又有直观的操作,学生很难在短时间内透彻掌握该课程众多的算法。为此,我们从兴趣激发和教学手段改进等方面积极探索提高理论教学效果的有效措施,通过优化案例和实验内容等加强实践训练来提高学生动手能力,引导学生运用数字图像处理的基本理论和算法思想解决实际应用问题。通过几年的努力,减少了学生的厌学情绪,加强了学生的实际操作能力和设计能力,提高了教学效果。

## 参考文献:

- [1]杨杰,黄朝兵. 数字图像处理及 MATLAB 实现[M]. 北京:电子工业出版社, 2014.
- [2]张永梅,马礼,何丽. 图像处理技术课程教学探讨与实践[J]. 计算机时代, 2014(10):76-77+80.
- [3]熊立伟,吴德华,薛云,张维. 地方本科院校“数字图像处理”课程教学改革思考[J]. 中国电力教育, 2014(03):107-108.
- [4]李洪均,蔡燕,张小美,张士兵. 以兴趣为导向的数字图像处理教学改革探索[J]. 中国教育技术装备, 2014(18):114-115.
- [5]阿里木江·卡斯木. “遥感数字图像处理”课程教学改革探析[J]. 吉林省教育学院学报(下旬), 2014(08):37-39.
- [6]肖龙飞,李金龙,杨凯,高晓蓉. 基于 MATLAB 的数字图像处理教学软件的设计[J]. 信息技术, 2014(12):185-187.
- [7]李娜,李大湘,毕萍. 《数字图像处理》课程的教学改革与探索[J]. 科技信息, 2014(05):38+51.

## 作者简介:

陈荣元(1976-),男,江苏兴化人,博士,高级实验师;  
刘星宝(1977-),男,山东临沂人,博士,讲师,本文通信作者;  
谭利娜(1981-),女,湖南长沙人,博士,讲师;  
申立智(1983-),男,湖南邵东人,硕士,助教;  
陈浪(1978-),女,湖南长沙人,硕士,助教。