

# 前端整合框架及在科研安全生产平台中的应用

张 睿<sup>1,2,3</sup>, 张 斌<sup>4</sup>

(1. 中国科学院 新疆理化技术研究所, 新疆 乌鲁木齐 830011; 2. 新疆民族  
语音语言信息处理重点实验室, 新疆 乌鲁木齐 830011; 3. 中国科学院大学, 北京 100049;  
4. 新疆维吾尔自治区产品质量监督检验研究院, 新疆 乌鲁木齐 830000)

**摘 要:** 为提高应用系统的前端性能及开发效率, 提出一种前端整合框架, 剖析其整合原理及整合的关键技术。以 AngularJS 为核心 js 框架, Node.js 作为基础环境, 集成 gulp、bower 及一系列插件, 融合组件化页面的前端开发思想, 具有自动化构建、模块化的目录结构和开发策略以及组件化开发过程的特点; 根据前端整合框架以及科研安全生产理论, 设计实现跨终端、基于角色的科研安全生产平台。实验结果表明, 前端整合框架在跨终端的、企业级的、任意规模的应用开发中具有广阔的应用前景。

**关键词:** 前端; 框架; 模块化; 自动化构建; 科研安全生产平台

**中图法分类号:** TP319 **文献标识号:** A **文章编号:** 1000-7024 (2017) 04-1086-07

**doi:** 10.16208/j.issn1000-7024.2017.04.045

## Front-end integration framework and application in research of safety production platform

ZHANG Rui<sup>1,2,3</sup>, ZHANG Bin<sup>4</sup>

(1. Xinjiang Technical Institute of Physics and Chemistry, Chinese Academy of Sciences, Urumqi 830011, China;  
2. Xinjiang Key Laboratory of Minority Speech and Language Information Processing, Urumqi 830011, China;  
3. University of Chinese Academy of Sciences, Beijing 100049, China; 4. Xinjiang Uygur Autonomous Region  
Product Quality Supervision and Inspection Institute, Urumqi 830000, China)

**Abstract:** To improve the reliability and development efficiency of the front-end of application system, a front-end integration framework was presented, and the principle and key techniques of integration were explained. Gulp, bower and a series of plugins, components of the front-end integration framework were designed in the light of componentized page, with the basic environment of Node.js and the core js framework of AngularJS. The characteristics of automation construction, modular directory structure and development strategy, and component development process were got. Based on the front-end integration framework and back-end technology, and in accordance with safety production theory of scientific research, a role-based, multi-terminal-applicable platform concerning safety production in scientific research was designed. Practice verifies that the front-end integration framework can be widely applied in application development in multiple terminals, at enterprise-level and of any scale.

**Key words:** front-end; framework; modularization; automation construction; research of safety production platform

## 0 引 言

通过应用前端整合框架替代开发人员完成前端开发中一些繁琐、机械性的工作, 增强前端代码的可复用性及稳定性, 提高前端性能, 并保证前端页面风格的统一性, 大

大提高了开发效率。前端框架的整合, 融合了物联网的思想, 为 PC 端和移动端的一体式开发提供了可能, 也同样增加了所开发平台的可移动性。本文在提出前端整合框架、分析其结构特性及整合原理的基础上, 完成了科研安全生产平台的开发工作, 该平台实现了危化品采购使用管理、

收稿日期: 2016-03-03; 修订日期: 2016-04-21

基金项目: 中科院院内部委级基金项目 (Y529401201)

作者简介: 张睿 (1990-), 男, 新疆乌鲁木齐人, 硕士研究生, 研究方向为计算机应用; 张斌 (1967-), 男, 新疆乌鲁木齐人, 高级工程师, 研究方向为计算机应用。E-mail: 895507981@qq.com

安全责任制、量化考核、巡检、隐患整改等业务功能。该平台前端完全基于前端整合框架进行开发,将前端开发的中心向业务进行转移,缩减了前端的开发、管理和维护工作。实际应用表明,前端整合框架在应用开发中具有广阔的应用前景。

## 1 前端整合框架

如今非常流行 Web2.0 和 HTML5 在很大程度上提高了 Web 应用的功能性和交互能力。越来越多的服务器端逻辑转移到了浏览器端来实现。浏览器的作用,已经从单纯的数据接收及展现演变成成为具有强大交互能力的应用载体。对于日渐复杂的浏览器端 HTML、CSS 以及 JavaScript 代码,需要一套更好的流程化工具来促进开发过程中各个环节能够协同、高效的进行。这其中也包括了前端代码全生命周期的管理,如初始的代码结构、开发流程、前端代码自动化测试等。

### 1.1 AngularJS

AngularJS 是一个 MV\* (Model-View-Whatever) JavaScript 框架,与传统的 MVC 架构例如 springmvc 从服务器端呈现的方式不同,Angular 的视图在浏览器端由它的模型以及所有的相关数据生成。控制器负责的协调 HTML 页面与模型之间的工作。Angular 无需服务器端干涉,所有工作都在客户端由缓存数据配合完成<sup>[1]</sup>。Angular 可自动对 DOM 节点进行操作,使前端开发重心向业务层转移。

### 1.2 AngularJS 五大特性

AngularJS 作为目前非常活跃的 JavaScript 框架之一,其作用之一便是要简化开发过程。因此,AngularJS 针对构

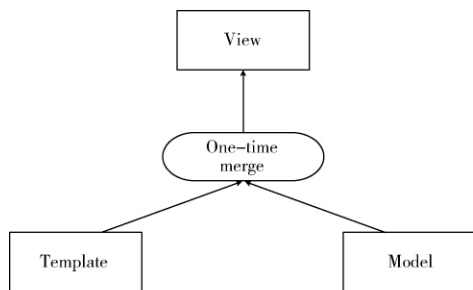
架小型 app 原型展现除了很大的用武之地。AngularJS 针对于功能性更加全面的客户端应用程序,其作用同样强大。AngularJS 的五大特性使得其具有了开发简便、特性广泛和出众的性能。

#### 1.2.1 良好的应用程序结构

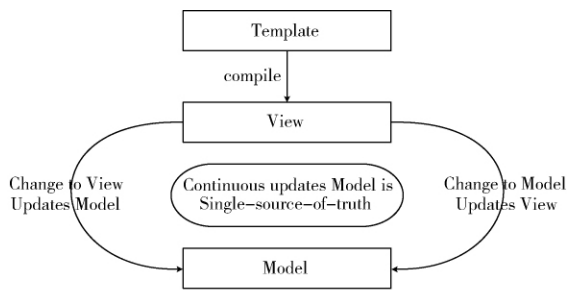
在使用原生的 JavaScript 或是 JQuery 进行大规模应用程序前端代码的开发过程中,由于它们缺乏明确的代码结构,随着代码量的增加,代码的维护成本和测试成本也同样在成倍的增加。AngularJS 的 MV\* 框架结构——MVC (模型-视图-控制器)、MVVM (模型-视图-视图模型)——使得开发过程中可以使用多种模式来组织源代码。AngularJS 的模块化使得每个模块的代码拥有自己独立的 scope (作用域),model (模型) 和 controller (控制器)。

#### 1.2.2 双向数据绑定

AngularJS 最吸引人的特性,便是它的双向数据绑定。通过 AngularJS,可以将模型与 HTML 中的元素进行绑定<sup>[3]</sup>。当模型发生变化时,视图会自动更新以同步模型的变化,实现单向数据绑定,如图 1 (a) 所示。同样的,在视图层发生变化时,数据的变化也将自动更新到模型层,即为双向数据绑定,如图 1 (b) 所示。在 AngularJS 的内部,通过 \$watch 和 \$digest 函数对被绑定的对象进行脏检查,如果发现模型或是 HTML 标签的值发生了变化,则进行更新操作。通过 \$apply 函数还可以在更新操作之后执行一个函数。AngularJS 的双向数据绑定,可以大大节省普通开发模式 (通过原生 JavaScript 或 JQuery 对前端应用进行开发,下同) 中针对 DOM 节点数据绑定的繁琐操作,从而使前端代码更加简洁、专一的实现业务逻辑而非系统实现。



(a) 单向数据绑定



(b) 双向数据绑定

图 1 单向数据绑定及双向数据绑定

#### 1.2.3 指令

Directives (指令) 的作用是引入新的 HTML 语法。指令是对 DOM 上的元素进行标记,使元素拥有特定的行为。AngularJS 原生自带了丰富的指令,比如数据绑定 (ng-model), DOM 循环 (ng-repeat), DOM 节点显示或隐藏 (ng-show、ng-hide、ng-if) 等。通过这些指令,可以赋予 DOM 元素特定的行为。指令的实现,是通过 DOM 元素绑定监听事件或者对 DOM 元素进行动态的改变而使 HT-

ML 拥有真正的交互性。

AngularJS 还支持自定义指令,一个 Angular 指令可以有 4 种表现形式: ①一个新的 HTML 元素 (<directive-name></directive-name>), ②元素的属性 (<input type = "text" directive-name/>), ③CSS class (<input type = "text" class = "directive-name" />), ④注释 (<!-- directive: directive-name -->)。这种使用组件的方式更加直接和清晰,可以轻易灵活了解插入 DOM 的指令元素的作用。

用,增加了代码的易读性和可复用性。随着开发工作的进行,组件库也在不断地完善和积累,这极大地减小了开发成本,也防止了开发人员使用大量的时间重复造轮子。

#### 1.2.4 HTML 模板

AngularJS 使用 HTML 模板,允许 UI 设计人员和开发人员协同工作。设计人员可以按照业务的需要设计和编写用户界面,开发人员则使用 AngularJS 的声明式绑定语法配合设计人员编写的用户界面绑定数据模型。这种方式在实现了前后端分离开发的前提下,更加细化地解耦了前端的开发工作,在后期的迭代工作中,用户界面或是数据模型的修改都有很高的可维护性。

#### 1.2.5 依赖注入和测试

依赖注入(dependency injection, DI)是一种软件设计模式,用于处理如何让程序获得其依赖(对象的)引用。对象或者函数有 3 种方式可以获取其依赖对象的引用:①依赖可以被使用者自己创建,通过 new 操作符。②依赖可以通过全局变量(如 window)来查找并引用。③依赖可以在需要的地方被传入。第①种和第②种创建或查找依赖使用的是硬编码的方式,因此它们不是最优的。采用这样的方式创建或查找依赖,会使得当依赖关系变得不可用时,需要修改依赖相关的代码才能修改整体的依赖关系,这种操作时非常困难和繁琐的。同样的问题也发生在测试中,因为测试通常需要通过模拟依赖来进行隔离测试。因此,第 3 种获取依赖引用的方式是最可行的,因为这种方式从组件中消除了查找依赖位置的任务,开发人员只需要将依赖传递给组件即可。AngularJS 便是选择了第 3 种方式来进行依赖注入,使用 injector 服务来创建和查找依赖,这个 injector 是一个 service locator(定位器)。例如我们需要注入一个 myModule 模块,只需 var injector = angular. injector(['myModule'])即可,AngularJS 会在页面初始化之前通过名称找到需要注入的模块,并将它注入进来。

AngularJS 的代码具有较高的可测试性,其 API 文档也列出了许多端到端的测试实例。其中,Karma 是 AngularJS 团队针对 JavaScript 开发的一个测试运行框架,它很方便地实现了自动执行测试任务,从而替代了例如回归测试集或是加载目标测试的依赖关系等繁琐的手工操作。开发者只需要在 Karma 中定义好配置文件来启动它,并在配置文件中制定相关的测试环境,Karma 便会在定义好的测试环境下自动执行测试用例。

## 2 前端自动化构建

如今的前端开发工作,已不再是简单的针对静态文件的编写工作了,针对于 Web App,前端代码的复杂度往往高于后端代码,针对第三方库的依赖、自动运行独立的前端测试、自动打包压缩发布代码等问题,传统的开发模式已经不能胜任了。因此,需要构建一个自动化的前端开发

流程,包括本地开发环境的初始化、第三方依赖的管理、源文件编译、自动化测试、发布到各个环境等。

### 2.1 gulp

gulp 是一个构建工具,开发者可以通过它在前端开发过程中自动执行配置好的任务。gulp 是基于 Node.js 构建的,gulp 的源文件和前端开发过程中用来定义任务的 gulp 文件都被写在了 JavaScript 或 CoffeeScript 里。前端开发人员还可以使用自己熟悉的语言编写 gulp 任务。gulp 本身不能完成自动化的任务,前端开发人员可以使用 gulp 的大量插件来完成预期的工作。

#### 2.2 选择 gulp 的原因

在 gulp 出现以前,前端自动化构建的解决方案非 grunt 莫属。grunt 是基于 Node.js 的自动化任务运行工具。grunt 有一个完善的社区,插件非常丰富。

##### 2.2.1 gulp 与 grunt 异同点

gulp 是一个构建工具,开发者可以通过它在前端开发过程中自动执行配置好的任务。gulp 是基于 Node.js 构建的,gulp 的源文件和前端开发过程中用来定义任务的 gulp 文件都被写在了 JavaScript 或 CoffeeScript 里。前端开发人员还可以使用自己熟悉的语言编写 gulp 任务。gulp 本身不能完成自动化的任务,前端开发人员可以使用 gulp 的大量插件来完成预期的工作。gulp 采用代码优于配置策略,grunt 遵循 CommonJS 规范,gulp 相比 grunt 更简洁。gulp 通过利用 Node.js 强大的流,不再需要向物理磁盘中写入中间暂存文件,可以更快的完成构建工作。gulp 很好地利用了流,这种通过管道将多个任务和操作连接起来的方式,使得所有的构建工作只有一次 I/O 过程。而 grunt 的 I/O 过程中会产生一些中间态的临时文件,在做自动或任务构建时,后边执行的任务会依据先前任务产生的临时文件进行后续的构建处理工作,并生成最终的构建后的文件。gulp 相比 grunt 更有设计感,其核心设计是基于 Unix 流的概念,通过管道连接操作,不需要写繁琐的配置文件。他们都有丰富的插件库,gulp 的每个插件只完成一个功能,这同样也是 Unix 的设计原则之一,其各个功能之间的整合是通过流的方式,并按照这种方式完成复杂的任务。

##### 2.2.2 gulp 的核心

gulp 是一个构建工具,开发者可以通过它在前端开发过程中自动执行配置好的任务。gulp 是基于 Node.js 构建的,gulp 的源文件和前端开发过程中用来定义任务的 gulp 文件都被写在了 JavaScript 或 CoffeeScript 里。前端开发人员还可以使用自己熟悉的语言编写 gulp 任务。gulp 本身不能完成自动化的任务,前端开发人员可以使用 gulp 的大量插件来完成预期的工作。

Node 中的 I/O 操作是异步的,因此磁盘的读写和网络操作都需要传递回调函数。其中流的种类有 Readable(可读)、Writable(可写)、Duplex(双工)及 Transform(运

算双工) 4 种形式。流可以是可读 (Readable) 或可写 (Writable), 或者兼具两者的。所有流都是 EventEmitter, 但它们也具有其它自定义的方法和属性, 取决于它们是 Readable、Writable 或 Duplex。

表 1 Stream 分类

使用情景	类	要实现的方法
只读	Readable	<code>_read</code>
只写	Writable	<code>_write</code>
读写	Duplex	<code>_read</code> , <code>_write</code>
操作被写入数据, 然后读出数据	Transform	<code>_transform</code> , <code>_flush</code>

Node 的 Stream 依赖 5 个模块运行。Liftoff 模块解决的问题是全局安装一个 CLI 工具, 但支持多个项目多个配置文件, 并且当前目录没有配置文件时, 可以就近向上级目录找到已有的配置文件, 或者在项目目录外执行命令行时可以指定配置文件的目录。所以 gulp 基于 liftoff 可以实现, 多个项目多个 Gulpfile, 并且可以执行 gulp 时指定配置文件路径。Through2 模块为 Node 的 streams2. Transform 的小型封装, 来避免子类的烦恼。可以更简单的通过一个函数来创建一个流, 而不用再繁琐的设置原型链的 `_transform`, `_flush`, 以及再扩充的 Transform 类中调用构造函数, 以便缓冲设定能够正确初始化。Vinyl 模块用来描述文件的一个非常简单的元信息对象, Vinyl 对象有两个主要的属性: path 和 content。它提供了一种简洁的描述文件的方式, 如果开发中需要访问本地文件系统上的一个文件, 还需要通过 Vinyl Adapter, 它会暴露一些方法: 如 `.src(globs)`, `.dest(folder)`, 和 `watch(globs, fn)`。globs 是路径模式匹配。Orchestrator 模块是一个基于 Node 的模块, 负责任务依赖关系定义, 处理和执行, 类似于目前所用的 AMD 模块加载器, 而且默认是最大限度的并行加载方式。

### 2.3 前端自动构建任务

前端自动化构建任务包括: ①自动加载。自动加载插件可以帮助开发人员自动加载 package.json 文件里的插件内容, 并且只在需要时才加载所需的插件, 避免工程冗余。重命名。使用 gulp-rename 用来重命名文件流中的文件。用 `gulp.dest()` 方法写入文件时, 文件名使用的是文件流中的文件名, 如果要想改变文件名, 可以在构建之前用 `gulp-rename` 插件来改变文件流中的文件名。②文件合并及压缩。在开发工作人, 由于协同开发的需要, js 文件和 css 文件会在多个页面上编写。在上线阶段, 需要将多个文件合并成一个文件, 帮助浏览器在一次请求后对页面进行缓存, 减少文件的网络请求, 提高页面速度。在合并之后使用对应的插件对 js、css、html 文件进行压缩, 更好提高访问效率。③语法检查。使用 jshint 插件检查 js 代码, 帮助开发

人员及时的找到代码中的问题, 提高开发效率。④图片压缩。使用 imagemin 插件对 jpg、png、gif 等格式的图片进行压缩。⑤自动刷新。使用 livereload 插件帮助开发人员自动刷新页面。当代码进行了修改后, 浏览器会自动刷新来展示改动后的效果。

前端自动化构建任务可为多个角色配置不同的任务。开发人员可以只针对自己所开发的模块进行语法检查, 由于 js 的压缩过程比较费时, 在开发测试阶段可禁止 js 压缩的任务执行, 而在项目上线时再进行压缩合并。美工人员可屏蔽 js 压缩合并及语法检查等任务的执行, 从而进一步更具不同角色的需求提高工作效率。

## 3 科研安全生产平台

科研安全生产平台由基础模块和业务模块两部分组成。项目的业务架构如图 2 所示。基础模块主要管理人员、部门、科研单位基本信息及与科研安全生产相关的政策法规。业务模块主要针对安全责任制的规范, 针对科研安全生产的量化考核, 针对科研人员的安全生产相关培训, 科研安全生产过程中的隐患整改、事故处置, 针对核化生爆危险品的进销存全生命周期管理以及针对单位重点部位、设施的日常巡更管理。项目是基于中国科学院 ARP3.0 系统研发的。系统截图如图 3 所示。

### 3.1 科研安全生产平台业务功能

广义上的科研生产安全项目应该包括 ARP 系统部分模块、管理协同工作环境、危险源全生命周期管理部分功能。3 大系统有机结合、互通形成了科研生产安全的大平台。从科研生产安全项目功能分布的角度上分析, 每个平台承载着不同的任务。管理协同工作环境的功能模块有: 巡检值班、责任制考核、事故报告、安全检查、宣教培训、重点部位管理等。ARP 系统提供的功能有: 预算编制、基础信息推送、网上报销等。危险源全生命周期管理提供的服务有: 危化品管理、放射与射线装置、生物安全等。

### 3.2 前端整合框架在科研安全生产平台中的应用

科研安全生产平台的主体思路是基于角色的管理平台, 该平台满足单页面应用的基本条件。由于该平台的业务框架相对复杂, 开发团队相对庞大, 开发周期较短, 需要在团队协作的前提下快速迭代开发完成。基于以上原因, 科研安全生产平台的前端开发任务是基于前端整合框架进行开发的, 使用了 AngularJS 作为前端框架, 并使用 node.js、gulp 为项目研发提供自动化的构建以及依赖管理依赖注入工作。针对科研安全生产平台短开发周期、多开发角色 (前后端开发人员、美工人员、UI 设计人员等) 同步开发的特点, 前端整合框架实现了针对不同开发角色的可定制的构架方式。

#### 3.2.1 可配置的语法检查

随着代码量的增加, 前端自动化构建的语法检查时间

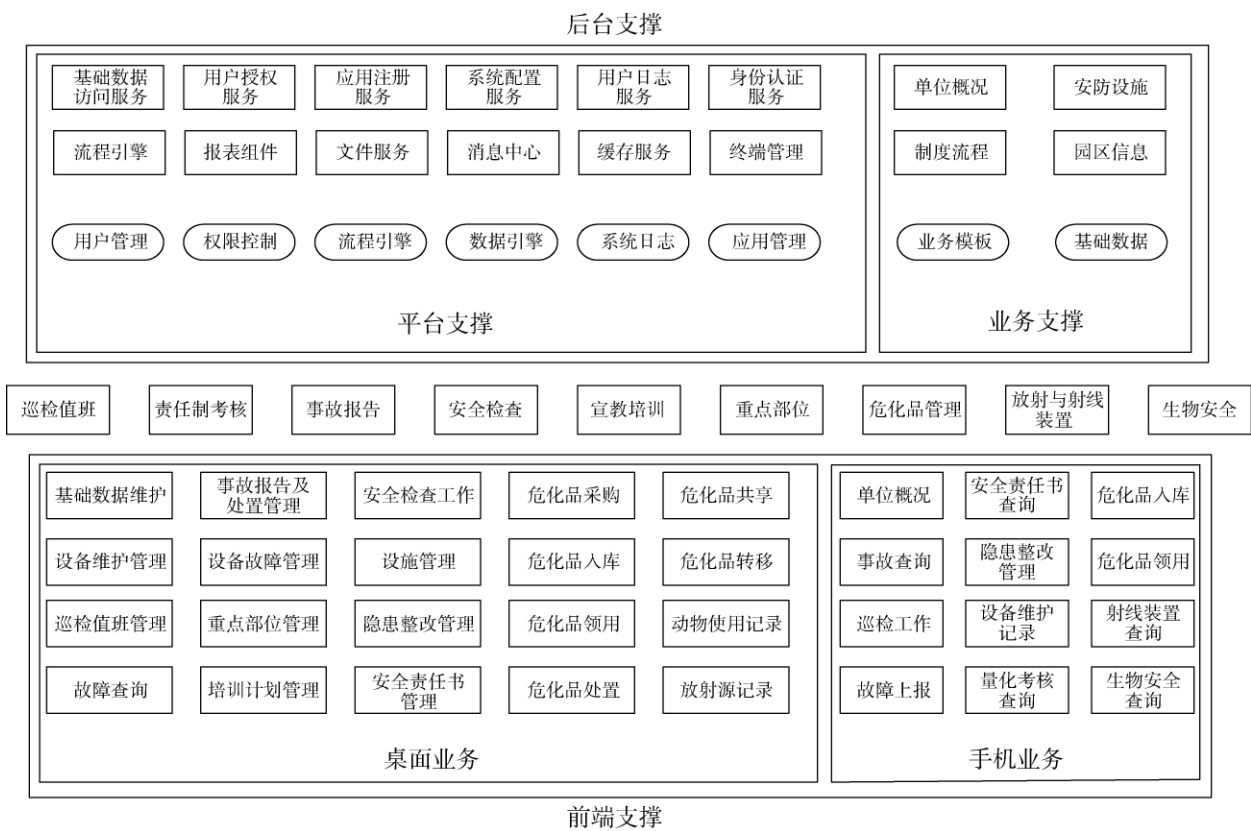


图 2 科研安全生产平台业务架构



图 3 科研安全生产平台系统

越来越不可忽略，大大影响了开发效率。项目采用团队开发的方式，针对整个前端代码的语法检查工作变得毫无意义，美工和 UI 人员更加不需要语法检查的工作。前端整合框架很好地解决了这个问题。开发人员可只对需要的文件或目录进行语法检查，美工及 UI 人员可关闭语法检查任务的执行。其核心配置代码如下：

```
if (withoutLint) {
  gutil.log('Ignoring \' + gutil.colors.cyan('lint') + '\'.');
  done();
}
```

```
} else {
  var jshint = require('gulp-jshint');
  var stylish = require('jshint-stylish');
  var htmlhint = require('gulp-htmlhint');
  return merge(
    gulp.src('src/' + lintDir + '/*.js')
  );
}
```

3.2.2 模块化的目录结构和开发策略

在传统的开发模式下，针对前端开发，一般采用按文件类型组织的目录结构。使用前端整合框架，可以按照模

块化的形式组织目录结构,与当前模块相关的所有类型的文件都可在同一目录下,避免了传统开发模式中 js 文件与 html 文件在不同的目录层级结构中,开发或修改时要花费时间去查找目录和文件的情况,提高开发效率。

通过前端整合框架,开发人员只需在前端定义自身页面与各个模块之间的依赖关系,通过 Controller 控制用户响应时需要载入的页面,这些页面会通过 AngularJS 封装的加载器在页面加载之前进行注入,加载器还会自动加载与这些页面有依赖关系的模块。模块化的开发策略如图 4 所示。

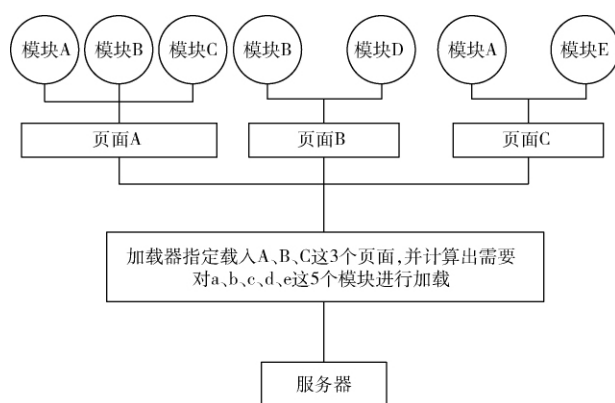


图 4 模块化的开发策略

### 3.2.3 组件化的开发过程

通过前端整合框架,根据科研安全生产平台的项目需要,已经积累了 30 多个前端组件,这其中包括了机构树、机构人员树、文件上传下载组件、地图组件、业务字典选择组件等等。在 pc 端的开发中,将常用的前端表现层页面封装成组件,例如表格表单、可收缩的手风琴样式等。开发人员只需使用一句 html 引入页面组件,并将数据按照特定的数据结构以 dom 属性的方式传给页面组件,即可完成一个前端页面的开发工作。在移动端也同样依赖前端整合框架组件库内的组件,通过 html 标签的引用,完成手机页面的开发工作。在这样的开发方式下,美工和 UI 人员只需要对组件进行设计和美化,并可保证设计美化工作可与开发工作同步进行。当页面样式需要调整或修改的时候,美工人员只需对页面组件进行修改,不再需要开发人员花费大量的时间重新进行套页面。这种开发方式,极大地减小了开发成本,缩短了开发周期,也避免了繁琐的机械化的操作。通过页面组件的形式进行开发,也保证了系统的风格统一性,对开发效率有着极大的改善。

### 3.3 基于角色的业务设计模型

科研安全生产平台使用基于角色的业务设计模型,跟传统的系统平台相比,基于角色的业务设计模型继承了 OA、ERP 的平台优点,克服了传统的以系统为中心,由用户去寻找系统中待办业务的模式,做到以用户为基准,不

同的用户为业务边界,待办业务由消息中心直接推送到指定用户。通过这样的机制,科研安全生产平台将用户分为了中科院总院领导,中科院分院领导,中科院总院下属研究院领导,安全管理人员以及科学院 5 个大类。针对不同的用户类型,提供了不同的体验方式。中科院总院领导在使用中侧重于数据的整合,平台首页主要为其展示各个下属单位的巡检执行情况、重点部位重点设施数量及具体信息介绍、是否存在安全隐患、安全隐患的整改情况、应急预案的建设情况等。平台为中科院分院领导提供针对该分院以及分院下属研究所科研安全生产情况的展示和操作功能。下属研究院领导可通过平台获取和操作该研究所的科研安全生产情况以及所涉及到的功能。安全管理人员操作具体的科研安全生产功能,如进行安全巡检、发起安全量化考核、进行危化品的采购工作等。科学家通过平台将日常的科研工作与平台相关功能结合,如危化品的申领、使用、销毁,安全责任书的签订,参与安全量化考核以及参加安全宣教培训等。

通过基于角色的业务设计模型,科研安全生产平台真正做到了针对每个用户的可配置的专有型平台系统,使得信息化的安全管理工作更加高效。当安全管理人员发现安全隐患或单位发生安全事故时,该消息将即时推送给相关领导和管理人员并根据具体推送内容通过不同的方式进行消息提醒(警报、普通消息的声音提醒等)。因为科研安全生产平台是建立在跨终端的基础上的,所以相关人员可及时的在 pc 端或手机端获取消息的内容,包括具体事项以及现场的照片视频等,在事故发生时,针对不同事故类型的应急预案也将推送到相关安全员手中,指引安全员正确的处理安全事故。

## 4 结束语

科研安全生产平台的开发工作已经顺利完成,得益于前端整合框架,系统的开发工作用时 60 天,经过试点单位的试运行,科研安全生产平台具有高可靠性、高扩展性的特点,页面风格统一,基于角色的业务设计模型为系统的使用带来了极高的友好性及易用性,为科研生产安全管理工作提供高效的解决方案。科研安全生产平台在短周期内研发的成功,很好地证明了前端整合框架是一个行之有效的高性能前端整合方案。前端整合框架用 AngularJS 作为 js 框架,将复杂的 dom 操作进行封装,将前端开发的中心向业务进行转移。自动化构建及组件化的页面开发模式,简化了前端的开发、管理和维护工作,大大提高了开发效率。前端整合框架在多终端的、企业级的、任意规模的应用开发中具有广阔的应用前景。

## 参考文献:

- [1] Balasubramanee V, Wimalasena C, Singh R, et al. Twitter

- bootstrap and AngularJS: Frontend frameworks to expedite science gateway development [C] //IEEE International Conference on CLUSTER Computing, 2013.
- [2] Richards G, Lebresne S, Burg B, et al. An analysis of the dynamic behavior of JavaScript programs [J]. ACM Sigplan Notices, 2010, 45 (6): 1-12.
- [3] Wilken J. Ionic in action: Hybrid mobile apps with ionic and AngularJS [M]. Manning Pubn, 2015: 100-120.
- [4] ZHONG Qiang. Research on Web front end architecture based on Node JS platform [J]. Wireless Internet Technology, 2013 (12): 124 (in Chinese). [钟强. Node JS 平台下 Web 前端架构的研究 [J]. 无线互联科技, 2013 (12): 124.]
- [5] DONG Yingru. A brief discussion on the application of AngularJS in the development of the next generation Web [J]. Software Engineer, 2015 (5): 30-31 (in Chinese). [董英茹. 简谈 AngularJS 在下一代 Web 开发中的应用 [J]. 软件工程师, 2015 (5): 30-31.]
- [6] CHEN Tao. Design and implementation of chat room based on AngularJS and Bootstrap [J]. Computer Knowledge and Technology, 2015 (9): 91-92 (in Chinese). [陈涛. 基于 AngularJS 和 Bootstrap 聊天室的设计与实现 [J]. 电脑知识与技术, 2015 (9): 91-92.]
- [7] Cantelon M, Holowaychuk TJ. Node js in action [M]. Manning, 2013: 37-67.
- [8] LI Rong. Research on Web application based on MVC model [J]. Software Guide, 2010, 9 (1): 19-21 (in Chinese). [李容. 基于 MVC 模式的 Web 应用研究 [J]. 软件导刊, 2010, 9 (1): 19-21.]
- [9] QIAO Shuyi. Research and implementation of key technologies of Web front-end framework based on MVC model [D]. Qingdao: Ocean University of China, 2014 (in Chinese). [乔淑夷. 基于 MVC 模式的 Web 前端框架关键技术研究 [D]. 青岛: 中国海洋大学, 2014.]
- [10] He Y, Luo M. Based on the ExtJS technology and SSH framework authority management research [C] //International Conference on E-Learning, 2010: 213-220.

(上接第 1044 页)

- [7] HE Xiaoyu. Unit commitment using an improved Lagrangian relaxation method [D]. Beijing: North China Electric Power University, 2009 (in Chinese). [何小宇. 基于改进的拉格朗日松弛法的机组组合优化方法 [D]. 北京: 华北电力大学, 2009.]
- [8] YU Jinsong, ZHANG Fan, WAN Jiuqing. Approach of optimal diagnosis tree generation based on dependence matrix [J]. Computer Measurement & Control, 2009, 17 (6): 1077-1083 (in Chinese). [于劲松, 张帆, 万九卿. 基于故障相关矩阵的最优序列生成方法 [J]. 计算机测量与控制, 2009, 17 (6): 1077-1083.]
- [9] HUANG Yifeng, JING Bo, LUO Binghai. Sequential multiple fault diagnosis strategy based on Rollout algorithm [J]. Control and Decision, 2015, 30 (3): 572-576 (in Chinese). [黄以锋, 景博, 罗炳海. 基于 Rollout 算法的序贯多故障诊断策略 [J]. 控制与决策, 2015, 30 (3): 572-576.]
- [10] SHAO Jiye. Study on model based fault diagnosis method and its application in aerospace [D]. Harbin: Harbin Institute of Technology, 2009 (in Chinese). [邵继业. 基于模型的故障诊断方法研究及其在航天器中的应用 [D]. 哈尔滨: 哈尔滨工业大学, 2009.]