

基于Axure的智能蚕房大屏数据可视化

庄晨怡

浙江理工大学, 浙江 杭州
Email: 594208284@qq.com

收稿日期: 2021年5月4日; 录用日期: 2021年6月2日; 发布日期: 2021年6月9日

摘要

智能化和现代化进程驱动下, 数据已经成为企业重要的生产材料之一。目前企业的大屏数据可视化需求日益增长, 已经成为目前可视化领域的一项热门应用。文中主要对大屏数据可视化系统搭建方法进行分析, 提出一种基于原型法的大屏数据可视化前端开发方法, 其主要基于原型开发软件Axure和Ajax技术框架, 并实现了一种不需要使用超文本标记语言进行大屏数据可视化搭建的系统, 为前端开发人员提供了便利。实践结果表明, 该系统成本低、易部署、可扩展, 能够快速实现前后端分离的大屏数据可视化系统开发。此外, 本文将搭建系统应用于蚕桑企业的数据展示, 打破传统模式中原型系统仅用于设计展示的局限, 在短时间内快速实现大屏数据可视化, 对相关工作的开发人员有参考价值。

关键词

大屏, 数据可视化, 原型法, Axure, Ajax, 前后端分离

Large-Screen Data Visualization of Intelligent Silkworm Room Based on Axure

Chenyi Zhuang

Zhejiang Sci-Tech University, Hangzhou Zhejiang
Email: 594208284@qq.com

Received: May 4th, 2021; accepted: Jun. 2nd, 2021; published: Jun. 9th, 2021

Abstract

Driven by the process of intelligent modernization, data has become one of the important produc-

tion materials for enterprises. At present, the demand for large-screen data visualization in enterprises is increasing, and it has become a popular application in the field of visualization. This paper mainly analyzes the construction method of large screen data visualization system, presents a method for front-end web developer of large screen data visualization based on prototyping. This system makes use of Prototyping software Axure and Ajax framework, implements a large screen data visualization system without using HTML, which can facilitate front-end developers. It has been shown that this system is low-cost, easy to deploy and scalable. It can be quick to set up and carry out with frontend-backend separation. Furthermore, this system is applied to sericulture enterprises for data presentation, which breaks limitation of the prototype system that only uses for design demonstration in traditional mode. This system can be built in a very short time. It is of reference value to the developers of related work.

Keywords

Large-Screen, Data Visualization, Prototyping, Axure, Ajax, Frontend-Backend Separation

Copyright © 2021 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

随着科技进步与发展,大数据时代的到来意味着数据已经在人们日常生活中占据越来越重要的地位。不同企业、不同领域都会产生各种各样类型的数据,但是这些数据往往是冗余且无法直接发挥作用价值的,现代技术常采用数据可视化来为人们提供一种易于理解的视觉感知形式。在此基础上以大面积屏幕为显示载体,从而实现数据展示的技术就叫做大屏数据可视化。大屏数据可视化通常借助图形设计、色彩选择以及布局排版等手段为用户提供良好的视觉体验,因此大屏数据可视化不仅能够在信息显示、数据分析以及监控预警等方面发挥重要作用,更是对企业文化的宣传与展现[1]。

近年来,由于蚕桑领域的现代化、智能化转型升级,生产环节中的数据统计工作量也随之增加,因此,越来越多企业在蚕桑养殖过程中使用现代信息技术来辅助生产。同时,借助大屏数据可视化技术将生产过程中产生的数据呈现在 LED 大屏幕上的企业需求也随之增多。大数据与可视化技术的结合带来了两个问题,一方面数据爆发式增长使海量动态数据的展示成为新难点,另一方面异构数据的多样性呈现方式增加了复杂性[2]。本文基于桑蚕养殖领域有关企业对智能蚕房大屏数据可视化的需求,采用原型法设计并实现了一种简单快捷的大屏数据可视化搭建方法,为蚕桑领域的智能化转型升级提供帮助。通过本文提出的方法能够省去传统模式下使用 HTML 语言进行网页开发的前端人员的工作,将 Axure 生成的页面直接与后台关联,实现数据传输与更新,从而大幅度减少大屏数据可视化系统的开发时间。

2. 相关研究

2.1. 原型法开发模式概述

大屏数据可视化技术是集科学技术与艺术设计美学于一体的综合性、跨学科技术。由于大屏制作设计环节基于企业需求,而企业客户对于大屏风格样式的需求又会受到企业文化、艺术素养、情感倾向、审美观念等因素的影响[3],因此采用原型法完成大屏数据可视化设计是比较科学高效的风格定义手段。原型是一个可实地运行的模型,有正式产品的主要特征,一般是软件系统的最初版本,软件原型能够以最少的费

用和最短的时间完成开发，并反映最终产品软件的主要特征[4]。使用原型法进行软件开发的模式符合大屏数据可视化对于界面美观的需求，能够在与用户沟通确定需要展现的大屏显示内容、确保设计的色彩风格符合用户喜好的基础上来进行大屏数据可视化的技术开发。同时，大屏数据可视化系统一般不存在复杂运算，处理过程较为简单，因此与原型法开发模式中适合处理过程明确、简单的小型系统这一特点相吻合。

原型法的基本思想是在限定的时间内用最经济的方法开发出一个可实际运行的系统模型。原型设计能够体现产品设计中的多元化和直观化[5]，需要尽可能的与用户产生联系。因此，原型法开发模式非常重视用户的项目参与度，用户能够在运行使用原型的基础上对其进行评价并提出修改意见，评价过程反复进行，直到满足用户需求。以原型为共同预研，实现用户与开发者的双向沟通，能够消除用户与开发组之间的隔阂，确保项目开发的软件与用户的需求规格相一致。因此原型法开发模式最大的优点在于能够满足用户的多样化需求，保证系统开发的实用性和可用性，从而降低开发风险。

传统的快速原型法开发模式过程一般大致需要经历需求分析与调研、原型设计、软件设计、开发编程、测试与投入使用六个步骤[6]。由分析人员与用户密切沟通，在用户的配合下迅速确定项目系统的基本需求，并将其特征用适合原型的方式描述，以满足开发原型需要[7]。完成分析评估后，使用专门软件对其进行原型设计，在基本需求的基础上尽快实现一个可行的系统。原型法开发模式中，原型设计阶段应重视开发者与用户的协调性，通过开发者与用户的充分沟通纠正交互中的误解和分析中的错误[8]，确保开发者制作的原型界面与运行效果能够满足用户对系统的需求后，将原型提供给软件开发人员参考，完成前后端开发编程，最后经测试再交予用户并投入使用。

2.2. 传统原型法开发模式的弊端

原型法遵循了人们对事物的认识是循序渐进和螺旋式上升这一客观规律，通过在设计环节对不同类型的原型进行试验和探讨，从而使产品更贴近用户需求。然而，以目前在软件开发编程环节项目开发的业界标准使用方式前后端分离模式为例，如图1所示为基于原型法的前后端分离开发模式流程，尽管原型法开发模式能在开发前期设计出符合用户需求的系统，为后续系统的可行性提供保障，从而节约测试成本，但其缺点也非常明显。原型法只能缩短用户与软件需求定义间的距离，并不能消除这个距离。传统的基于原型法的软件开发模式往往只将原型作为与用户沟通交流的工具，原型系统得到用户的确认后一般就会被抛弃。尤其是大屏数据可视化一类的软件开发，借助 Axure 等原型设计软件制作的原型只能确定项目的功能需求、确定设计的色彩风格等开发人员仍需重新进行开发编程，因此，原型系统无法提供技术支持、无法从技术层面为开发人员减轻负担，前端开发人员的工作量依然非常庞大，其不适合大型、复杂项目的开发。

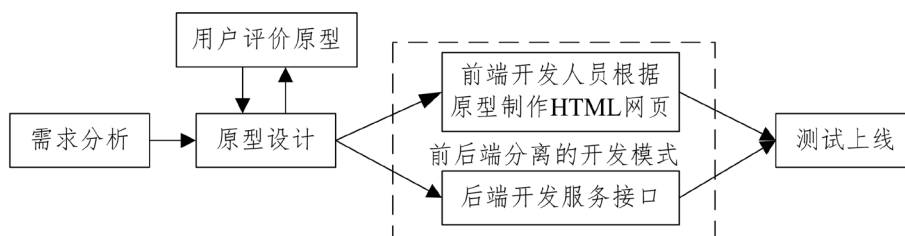


Figure 1. Flowchart of frontend-backend separation development mode based on prototyping method

图 1. 基于原型法的前后端分离开发模式流程图

原型开发工具 Axure 作为业内常用的产品原型交互软件，Axure 能以高保真的 WEB 及移动端原型与用户进行沟通和交流，是目前主流的交互原型设计软件，在操作难度、设计效率等方面具有很大优势[9]。然而 Axure 软件自动生成的基于 HTML、JavaScript 和 CSS 的网页一般是静态页面，只能完成简单的交

互。且由于自动生成的网页代码可阅读性差，由 Axure 软件自动生成的网页只能为后期开发工作中前端人员提供浏览功能，前端开发人员一般不会在原型 HTML 代码上进行修改或优化。这表明原型设计阶段生成的系统雏形无法参与后续前后端分离的开发环节，前端开发人员仍需从零开始进行网页代码的编写，会对前端开发人员的工作量造成极大负担，使项目的开发成本大幅提高、开发效率显著降低。

因此，本文对传统原型法进行改进，提出一种能够在 Axure 中完成完整的前端开发，使 Axure 生成的页面能够完成与后台和服务器的数据交换，而不仅仅是一个展示用的静态文档，在 Axure 中引入 Ajax 技术框架使网页实现异步更新，在不重新加载整个网页的情况下对网页的某部分进行更新[10]。本文提出的基于 Axure 的大屏数据可视化方法不再需要前端开发人员对 Axure 生成的 HTML 网页进行重构或优化修改，且对后台的逻辑要求极低，只要求后端提供与前端一致的数据接口。

3. Axure 中数据可视化大屏的设计与实现

3.1. 系统分析与设计

大屏数据可视化系统一般基于企业需求，能够针对性的展现企业经营环节、生产环节、运营环节等一个或多个环节中产生的数据。本文所设计的智能蚕房大屏数据可视化系统针对蚕桑企业生产环节产生的数据，不仅需要满足用户对美的多样性需求，还要对企业生产环节的客观数据进行呈现，同时提供工作安排的展示功能，从而更好的服务于生产环节相关员工的工作进程。此外，在大屏可视化设计阶段的排版布局、图标选用等应服务于业务[11]，根据客户的需求增加新颖的图形、图表供客户选择。

经调研分析后了解到，如果从蚕桑角度解释生产环节内容，大规模的蚕桑养殖一般将养殖过程分为小蚕房、中蚕房、大蚕房三个种类，各个规模的房间数量不一，且各个房间具有各自的实时温湿度、设定温湿度、设定误差值等多个数据信息。因此，蚕桑领域的智能化大屏数据可视化系统开发需要综合蚕桑养殖环节产生的多种数据信息，应以房间为单位，在系统中应当注重不同房间之间的独立性、明确房间与房间自身数据的对应关系。然而从人员角度解释对生产环节工作进行分析时，相关养殖人员的工作内容则不再以房间为单位，而是基于规模。例如，小蚕房一般多个房间位于同一建筑，因此该建筑内的所有小蚕房房间的工作内容安排一致；而大蚕房由于规模大，一般一个大蚕房为一整栋建筑，因此多间大蚕房对应多个工作安排。因此，如图 2 所示，在系统设计时应当将大屏数据可视化分为房间温湿度数据展示功能模块与工作安排展示功能模块两部分，针对各自的数据特点实现数据展示与定时轮播。

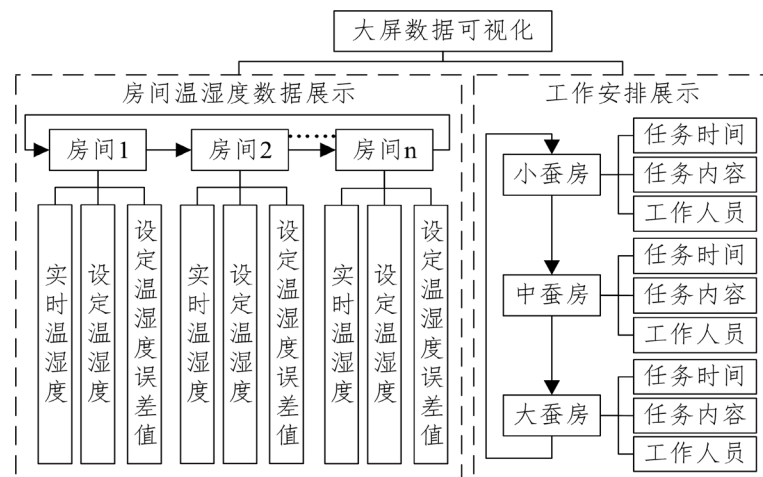


Figure 2. System architecture of large-screen data visualization of intelligent silkworm room

图 2. 智能蚕房大屏数据可视化的系统架构

本系统采用动态大屏数据可视化进行展示，能更好的帮助用户理解数据信息内容、观察数据的对比性。另外，为提高大屏的可观赏性，将数据信息按照类别、时间、位置等层级进行合理组织，并且采用数字、文字、形状、图表等多种数据表现形式来呈现数据，从而使用户有能够在恰当的时间间隔中获取复杂数据信息。

3.2. 系统开发与实现

针对大屏数据可视化所需要表现出的艺术性与观赏性，在软件开发开始前需要对大屏数据可视化的界面进行设计。与传统的原型法软件开发模式相同，首先需要借助原型设计软件针对业务需要对大屏的可视化元素进行形状图案、色彩风格以及整体布局的设计。

本文所投放的大屏类型为 LED 无缝隙大屏，因此采用和电脑显示器一致的分辨率，即 1920*1080。如前文所述，原型设计最重要的特点在于重视用户的参与度，因此在与用户不断的沟通协商、明确用户对界面的需求后，完成大屏的初步设计并交予用户确认。本文使用常用的原型设计软件 Axure 完成原型的设计，根据系统功能和用户需求设计的界面布局线框图如图 3 所示。

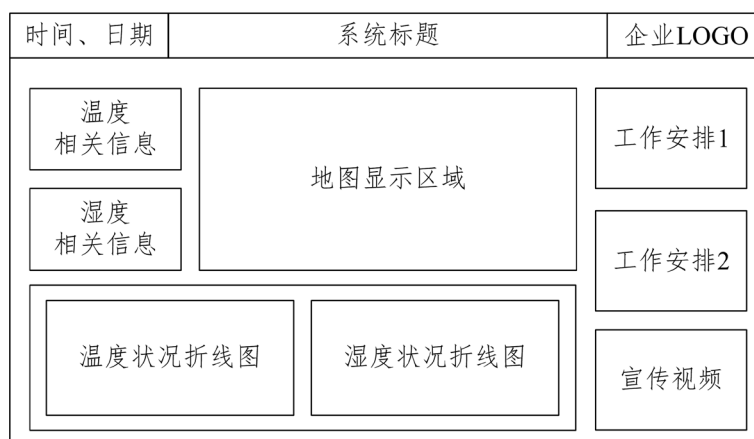


Figure 3. Wireframe of interface layout

图 3. 界面布局线框图

大屏数据可视化的界面设计完成后需要在 Axure 中实现基础交互功能。Axure 提供用于定义结构和逻辑的接口，在设计交互功能时能够消除对原型进行编码的需要。Axure 中交互功能的实现以事件为单位进行。不同事件存在不同的触发条件，每个事件可以包含多个场景，同时一个场景可以执行多个动作。利用 Axure 提供的可视化工作环境能够简单快速的以鼠标方式在线框图上定义连接和高级交互，从而实现页面的动态展示效果。

基于系统分析与设计环节的流程架构，针对蚕桑养殖过程的特点，首先应在大屏数据可视化系统中设计多个房间，为每个房间的实时数据进行展示。基于动态数据可视化设计的理论，在可视化展示时应当突破静态屏幕、连续时间的观念，遵循“注意力捕捉”的视觉模态认知概念规则，使时间轴运动起来[2]。使用 Axure 的 Panel 面板控件，在面板中为每一个房间制作一个面板页，从而保证系统中不同房间的独立性。以实时温度这一数据为例，大屏中需要呈现的实时温度数据的数量显然是基于房间数量的，本文开发的大屏数据可视化系统需要呈现 8 个房间，因此每次只需要了解到 8 个实时数据信息，使用大小为 8 的数组以数组形式存储不同房间对应的实时数据，同时引入一个全局变量对当前大屏所展示的房间进行下标的记录，就能保证房间与房间自身数据的对应关系。

使用 Axure 的 PANELSTATECHANGED 事件实现蚕房房间轮播效果，房间跳转时显示数据根据房

间切换内容的流程思路如图 4 所示。面板固定频率自动切换到下一个房间所在的面板，每次切换房间时获取一次当前房间名，从而确定一个唯一的 `current_room` 值。针对智能蚕房系统的不同房间，每个房间对应有自己的实时温湿度、设定温湿度以及设定的温湿度误差值，每次切换面板时都会将 `current_room` 作为下标值，从对应网页全局变量中获得指定下标位置的数字，并更新控件内容，实现每个房间温湿度数据的展示。

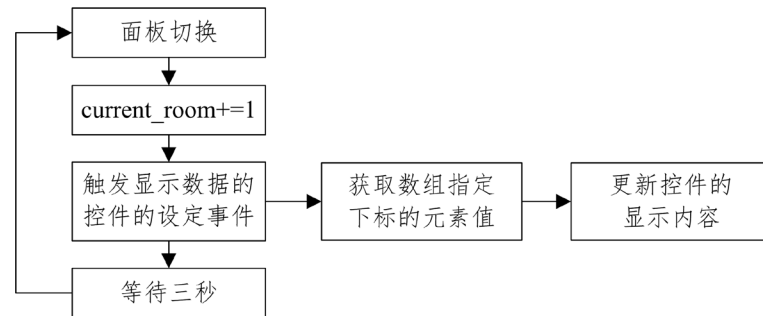


Figure 4. Flowchart of invoking PANELSTATECHANGED event

图 4. PANELSTATECHANGED 事件调用流程图

类似的，工作安排展示部分也采用轮播形式。由于工作内容不再根据房间而是基于蚕房规模，设计小蚕房、中蚕房和大蚕房三类，每个部分都包含各自的任务时间数组、任务内容数组以及工作人员数组。这三个数组相同下标位置的内容都是一一对应的，存在映射关系。

上述运行流程仅仅完成了静态网页的开发，还需要在 Axure 中使用动态网页技术，在后台与服务器进行数据交换，才能真正满足大屏数据可视化的使用需求。由于各个控件都是基于全局变量实现数据展示的，因此考虑通过修改全局变量来实现数据实时更新的功能。同时，大屏数据可视化一般只需要呈现在 LED 大屏上作为展示，很少需要用户点击或鼠标交互，因此采用 Ajax 技术，在不更新整个页面的前提下维护数据比较合适。Ajax 技术打破页面重复加载的惯例，通过异步操作获取服务器数据，同时它的代码执行和数据处理都在后端进行，能够减轻前端的负担，能够为用户创建和实现更动态、更丰富的软件界面[12]，这正与大屏数据可视化技术的开发需要相吻合。本文设计的系统在页面加载事件中执行 Link to external URL 类型的 Open Link 动作，在该动作中可以使用 Ajax 来与后端进行数据获取。

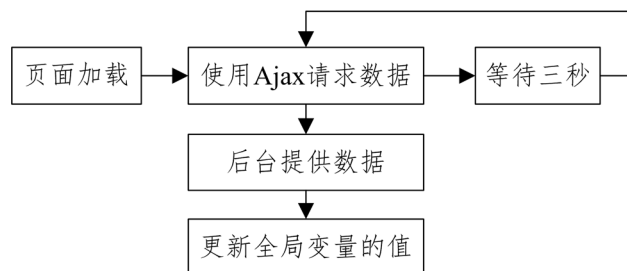


Figure 5. Flowchart of page load events using Ajax technology

图 5. 页面加载事件中使用 Ajax 技术的流程图

如图 5 所示，在页面加载事件中执行 Ajax 代码向后端发起 POST 请求，使用后端返回的数据修改全局变量。同时，页面加载事件中经过三秒的等待时间后会进行自刷新，这意味着每三秒就由 WEB 服务器提交一次 POST 请求，实时修改全局变量的值。如前文所述，网页中控件显示的数据是基于全局变量的，每次执行 Ajax 代码先后台请求数据并更新全局变量的值后，就能使前端网页在不重新加载页面的基

础上将控件上显示的数据进行修改,从而实现数据展示的实时性。

在 Axure 中写入的 Ajax 代码格式与传统前端开发方式中使用的 JQuery 提供的 Ajax 方法类似,可以指定 method 为 POST 或 GET 等方法,在 url 中指定所需要请求的后端服务器地址,使用 Ajax 特定的事件处理函数如 success 方法、error 方法对执行终止时的事件方法进行设计。在 Axure 的事件中使用 Link to external URL 类型的 Open Link 动作需要输入的代码格式如下:

```
javascript:
$.ajax({
  method: "/*请求方法*/",
  url: "/*请求地址*/"
}).done(function(msg){
  $axure.setGlobalVariable("/*全局变量*/", JSON.stringify(msg["/*返回数据*/"]));
});
```

本文采用 Ajax 中与 success 方法有类似功效的 done 方法,当 Ajax 事件成功处理后执行该方法,利用 Axure 中提供的 setGlobalVariable()方法将 Ajax 请求后台获得的返回数据写入全局变量。采用 Ajax 技术框架能够提高网页的加载速度、保证程序的简单性,同时方便用户处理 HTML [13],这在本文需要搭建的高并发前后端数据交换环境中对运行效率有很大提高。

本文设计的系统在后端开发环节采用 SpringBoot 框架,选取 SQLServer 数据库。使用上述流程生成的网页中已经通过 Ajax 技术实现了向后端的数据请求,且请求频率为每五秒一次。后端接收到网页客户端发起的 POST 请求后根据请求信息查询数据库,将查询到的数据以符合前端数据接口要求的格式存入 request,由服务器转发回网页客户端。采用本文提出的大屏数据可视化方法对后端代码的逻辑要求极低,后台使用的框架技术对前端来说具有透明性,即只要后台返回的数据格式与前端所需要的严格一致即可。本文提出的改进方案保留前后端分离开发模式的优势,前后端之间只需要进行数据传输[14],后端开发人员不需要考虑前端页面代码组织,提高了后台开发人员工作的易操作性。

Table 1. Part of the global variable information in Axure table

表 1. Axure 中部分全局变量信息表

Variable Name	Variable Value	Meaning
temperature	["26.2", "25.7", "25.2", "24.7", "26.1", "25.5", "28.1", "24.9"]	Real-time temperature
tpPrest	["26.0", "25.0", "24.5", "24.0", "25.5", "24.0", "26.0", "26.0"]	temperature setpoint
tpErrot	["1.0", "2.0", "3.0", "4.0", "1.0", "2.0", "3.0", "2.0"]	temperature error
plan	["喂食", "消毒", "清扫", "喂食"]	job content
time	["06000900", "11001200", "16001700", "17301830"]	working time
worker	["张三", "王五", "李四", "王五"]	staff

必须要注意的是,由于 Axure 对数组的操作存在局限性,采用下标对全局变量进行获取时必须严格控制数据格式,必须确保后端返回的数组中每个下标的字符串长度符合各个控件所需要获取的下标位置,才能保证最后大屏呈现出的数据正确。因此,文本所提出的基于 Axure 的大屏数据可视化方法在开发环节要求前后端所使用的数据接口严格统一。同时,考虑到后端返回的数据内容的特点, Axure 中的变量值中应当添加方括号。其次,为方便后端控制数据长度,将所有的数据类型统一为字符串类型,在设计变量值时应考虑到字符串类型的数据和数值型数据在计算机中的表示格式的区别,数组中的每个变量都

需要以双引号括起。以本文实现的大屏数据可视化系统为例，在 Axure 中所使用的全局变量中部分变量信息如表 1 所列。

以 temperature 为例，语法格式为 $\text{substr}(2+(\text{current_room}-1)*7,4)$ ，其中第一个参数表示从字符串的哪个位置开始，第二个参数表示需要获取的字符串长度，可见对应控件从该变量中取值的范围依赖于下标。当下标 current_room 值为 1 时，从第 2 个字符开始获取，这是考虑了方括号和双引号在字符串中的影响；在 temperature 中每一个数组成员都有 4 位有效数据信息、2 个双引号以及 1 个逗号数组分隔符，因此需要以 7 为倍数；有效数据长度为 4，因此第二个参数大小设置为 4。基于上述分析，采用本文提出的大屏数据可视化方法时，前后端应当对数据格式进行多次确认和交流。使用 Axure 开发前端的工作人员设计的全局变量格式时，应当考虑到后端返回的数据会是何种格式，后端也应当严格按照前端提供的数据接口实现请求响应，并且在返回数据前对每个下标的数据长度进行校验，才能防止网页中出现数据错位的问题。

4. 测试与评估

开发完成后软件系统运行界面如图 6 所示。大屏正中央的区域为地图区域，采用不同位置或大小的矩形对不同房间进行区分，并以地图定位气泡表示当前显示数据信息对应的具体房间。地图左侧的两个模块分别实现每个房间自身六个数据的信息展示。底部的折线图引入 highcharts 技术框架动态展示每个房间对应的十二小时内的实时温湿度信息，以图表形式直观的对该房间的温湿度信息进行分析与展示。右侧三个模块实现工作情况的展示以及企业宣传视频的播放。当后台数据变化时，可视化界面中的曲线、数据实时发生变化。

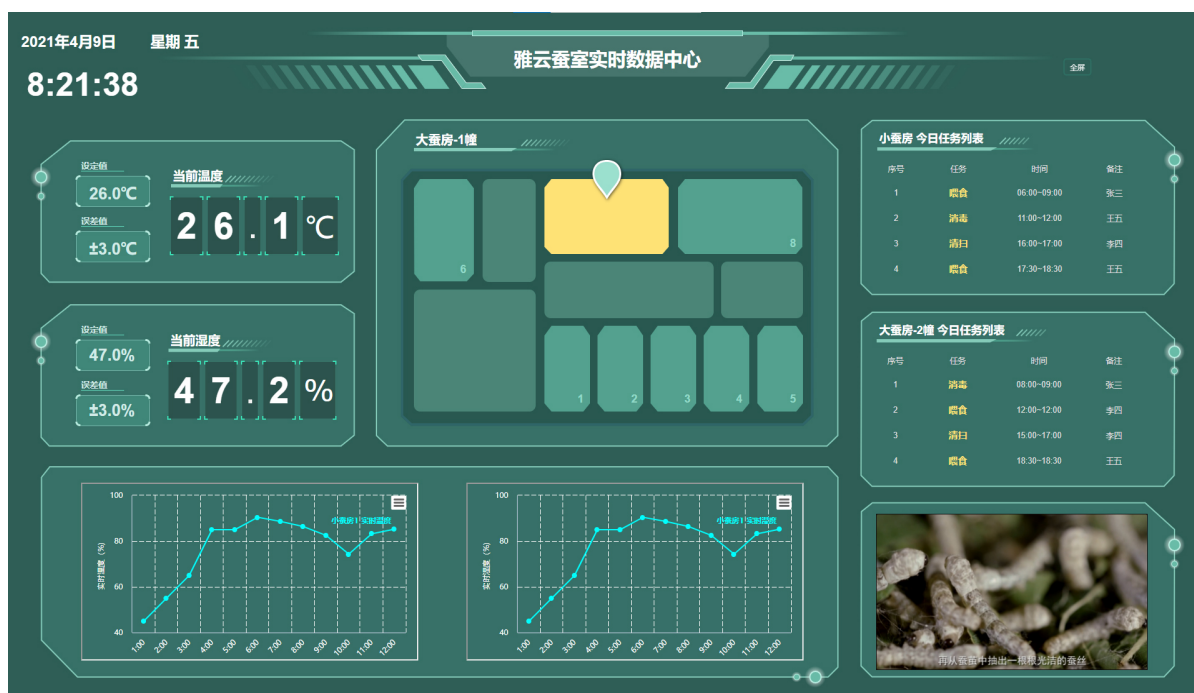


Figure 6. Flowchart of page load events using Ajax technology

图 6. 页面加载事件中使用 Ajax 技术的流程图

测试期间，软件执行其功能时，响应和处理时间及资源消耗均在可接受范围内；操作界面简洁易懂，各种信息易于理解；测试期间未出现数据丢失、系统紊乱或死机现象，具有较好的稳定性和安全性。

造成 Axure 在传统原型法中的局限性的原因之一正是其自动生成的 JavaScript 文件可读性差,导致前端开发人员不能对其直接进行修改。因此,本文提出的基于 Axure 的智能蚕房大屏数据可视化技术在传统的基于原型法的前后端分离软件开发模式上进行改进和创新,采用在 Link to external URL 类型的 Open Link 事件中执行 Ajax 代码的方法,能够在 Axure 自动生成的 JavaScript 文件中直接将使用 Ajax 技术框架的 JavaScript 代码构建在其中,从而实现与后端的连接。如图 7 所示为本文提出的优化的基于原型法前后端分离开发模式流程。将 Axure 这一原型开发工具参与到前后端分离的开发环节中,充分利用 Axure 自动生成的网页包含 HTML、CSS、JS 等网页渲染必要组成部分的这一特点,将 Ajax 技术框架融合到 Axure 中,减去了前端开发人员根据原型开发 HTML 页面一步,相较于传统的基于原型法前后端分离模式开发过程更加快速简单。同时,保留前后端分离开发模式,将前后端代码进行解耦,使得开发人员能够各司其职,前端使用 Axure 专注页面表现,后端专注高性能和业务,两者并发进行,保证项目的可维护性和可扩展性、提高开发效率。

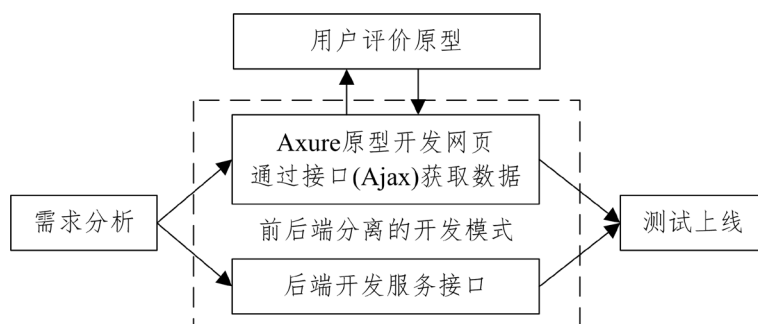


Figure 7. Optimized flowchart of frontend-backend separation development mode based on prototyping method

图 7. 基于原型法的前后端分离模式开发流程图

采用本文提出的基于原型法的大屏数据可视化方法完成的系统兼顾界面的美观性与功能的全面性,能够很好的将相对抽象的复杂数据以易理解的可视化方式给用户带来良好的体验与表达,从而辅助蚕桑养殖领域相关人员的工作,建立起用户与数据之间的辅助决策枢纽[3] [5],同时也是对企业文化的良好宣传与展示,最终实现数据与商业之间的完美结合。

5. 结语

目前越来越多的应用软件或项目开发采用基于原型法的软件开发模式,并且这种开发模式在项目开发过程中的优势越来越明显,能够为项目提供良好的基础。本文提出的由 Axure 完成前端向后端获取数据的方法,能够直接省去原型法开发模式中前端开发的工作,为有搭建大屏数据可视化系统要求的团队和前端开发人员减轻工作负担,使大屏数据可视化开发变得更高效率,进一步缩短大屏数据可视化的开发时间。这种综合利用 Axure 与 Ajax 技术的搭建技巧能够使 Axure 自动生成的网页文件具备与后端服务器进行数据交换的功能,不再局限于原型设计,克服传统的原型设计模式的不足,突破 Axure 仅作为原型设计软件的局限性。此外,智能蚕房大屏数据可视化系统是将图形图像设计与数据统计分析有机结合的一种理念技术,同时也是实现蚕桑行业现代化、智能化、分析科学化和动态化的重要组成部分,因此具有广阔的发展前景。

由于 Axure 本身能实现的功能存在限制,尽管采用本文的方法能实现智能蚕房大屏数据可视化,但本方法仍不适用于一些有复杂功能需求的软件开发项目,仅能为有快速搭建大屏数据可视化需要的开发人员提供一种高效实现方式思路作为参考。是否能在复杂网站系统开发中使用本文提出的方法,还需要

在以后的研究中进一步探讨并优化。

致 谢

感谢在整个论文写作过程中帮助过我的每一位老师和同学。首先,也是最主要感谢的是我的指导老师,在整个项目完成过程中给予了我很多帮助,教导了我很多。在开发过程中我也遇到了问题,思路不清晰时,经过老师的点拨,最终完成了项目工程。在论文写作方面,完成初稿后,老师认真查看我的文章,指出了我存在的很多问题。在此十分感谢老师的细心指导,才能让我顺利完成项目开发和论文撰写。其次,感谢在项目开发过程中一起合作的同学们,是我们共同的努力才使得项目能够顺利完成,在此也十分感谢他们的付出。

参考文献

- [1] 李金金. 大屏数据可视化系统中的视觉表现探析[J]. 大众标准化, 2020(16): 151-153.
- [2] 吕月米, 周雨. 基于多模态理论的大数据可视化的优化与拓展[J]. 包装工程, 2019, 40(24): 251-259.
- [3] 张开智, 万向, 陈亚民. 关于数据大屏可视化的设计流程与发展趋势的分析研究[J]. 科技创新导报, 2019, 16(27): 130-133.
- [4] 姜文, 刘立康. 软件原型系统在软件项目开发中的应用[J]. 计算机技术与发展, 2019, 29(4): 110-115.
- [5] 苏思远. 工程现场管理 APP 的用户体验设计[D]. [硕士学位论文]. 北京: 首都经济贸易大学, 2017.
- [6] 董元和, 张行文. 基于 Axure 的软件原型法的研究与应用[J]. 湖北师范大学学报(自然科学版), 2017, 37(4): 15-19.
- [7] 胡伟华. 面向企业的银行资金归集平台的设计与实现[D]. [硕士学位论文]. 杭州: 浙江工商大学, 2014.
- [8] 邵中惠. 软件开发中的原型化方法[J]. 淮南职业技术学院学报, 2005(3): 93-96.
- [9] 于红艳. 基于 Axure RP 的智能审稿专家遴选系统原型设计[J]. 中国科技期刊研究, 2019, 30(9): 976-982.
- [10] 宋文文, 孙力群. 大数据可视化数据加载模式比较分析[J]. 电脑知识与技术, 2019, 15(36): 11-12.
- [11] 李井泉, 刘惠颖. 快速原型法在数据中心中应用[J]. 计算机与信息技术, 2010(4): 75-76.
- [12] 吕振刚. 基于 Ajax 技术的 WEB 研究[J]. 石油知识, 2020(4): 50-51.
- [13] 王艳梅. jQuery 在 AJAX 技术框架中页面加载的应用分析[J]. 电脑知识与技术, 2020, 16(36): 210-212.
- [14] 孙思杰. Web 项目基于前后端分离模式的设计与应用[J]. 科技创新与应用, 2020(27): 96-97.