|  |
| --- |
| 软件工程实验I组 |
| 软件项目计划书 |
| 对Node-RED工具的消息聚集和图形可视化拓展 |

|  |
| --- |
| 暴明坤、夏欣怡、胡俊涛、张雨濛、叶柏威、刘子渊  2020-3-18 |

目录

[一、 引言 2](#_Toc36046067)

[1.1 项目背景 2](#_Toc36046068)

[1.2 项目目标 3](#_Toc36046069)

[1.3 术语定义 3](#_Toc36046070)

[1.4 参考资料 4](#_Toc36046071)

[1.5 标准、条约和规定 4](#_Toc36046072)

[二、 项目概述 4](#_Toc36046073)

[2.1项目目标 4](#_Toc36046074)

[2.2项目可交付内容 5](#_Toc36046075)

[2.3工作内容 5](#_Toc36046076)

[2.4项目及软件工程综合实验应交付成果 5](#_Toc36046077)

[2.5项目验收方式和依据 6](#_Toc36046078)

[三、 项目团队组织 6](#_Toc36046079)

[3.1项目结构 6](#_Toc36046080)

[3.2组织划分 7](#_Toc36046081)

[3.3人员分工 7](#_Toc36046082)

[3.4协作与沟通 7](#_Toc36046083)

[四、 实施计划 8](#_Toc36046084)

[4.1计划分析 8](#_Toc36046085)

[4.2总体进度计划 8](#_Toc36046086)

[4.3各小组进度计划 9](#_Toc36046087)

[五、 关键问题 10](#_Toc36046088)

# 引言

## 项目背景

物联网与Node-RED

**物联网（Internet of Things，简称IoT）是上个世纪就已经开始发展的互联网趋势，其主要目标是让尽可能多的设备连入互联网，让更多的数据通过网络连接流动，而不是低效的人-人、人-电脑传递。进入21世纪之后，随着硬件成本的降低和性能功耗比的提升，物联网的研究和应用才逐渐推广开来，例如智能家居也是最近几年才走进大众家庭中。而最近5G技术的突飞猛进，扫平了万物互联在网络上的障碍，进一步推动了物联网开发的热潮。**

**在物联网开发过程中的一个核心问题是传统命令式编程（Imperative Programming）不能适应需求的复杂性和灵活性，需求变动带来的是重新组织、重新编译、重新部署程序带来的低效性，而文本代码形式的编写方式也不能够直观地展现数据处理的具体流程。因此在物联网领域很早就先行的IBM公司带来了基于流编程（Flow-Based Programming）范式的事件驱动（event-driven）流程编排工具Node-RED，其将每个数据处理的步骤抽象为节点（node），通过拖拽创建节点、改变节点位置，以及连线表示节点之间的数据流动方向，给用户以方便且动态地编排并部署工作流（flow）的可能，而部署后的工作流则由事件触发执行。Node-RED工具分为前端和后端，均由JavaScript开发，前端通过网页提供交互式编排的功能，并将部署的工作流信息发送到后端，由后端实例化工作流，监听事件并触发流的执行，同时可以在节点执行过程中一些信息通过WebSocket返回给前端，以提供监控的功能。**

**Node-RED工具另一个顺应物联网开发需求的特性是模块化，其通过良好的代码结构设计实现了节点、流程与工具本身的解耦，并集成了可供开发人员共享节点和流程的社区，因此用户可以一键下载并热插拔节点和流程，尽可能地复用他人已经实现的数据处理需求。**

Node-RED在国内的现状

**Node-RED的国外社区较为繁荣，但一些实用功能并没有国内开发者进行本土化，比如消息聚集拓展包可以通过即时通信软件如Telegram、Facebook Messenger、Slack等提供的机器人机制，将流程执行中的信息实时主动地反馈给用户，也可以让用户从移动端发消息给Node-RED作为事件来触发流程的执行，极大地拓展和丰富了Node-RED的使用场景。而国内即时通信软件，如阿里的钉钉，字节跳动的飞书等均未被Node-RED支持。**

## ****项目目标****

本组计划借助本土办公通信软件例如飞书或钉钉等，开发Node-RED工具的本土化消息聚集拓展包，方便国内物联网开发人员的使用，填补Node-RED在中国本土化的空白。除了本土化消息聚集扩展包外，为了推送消息的图文并茂，本组亦计划开发一个将数据绘制为图表和进行Markdown模板渲染的可视化拓展包，并且最终通过一些使用场景示例的搭建展示这两个拓展包的实用价值。

## 术语定义

关于本项目中的一些专业术语如下表1所示。

**表 1 专业术语表**

|  |  |
| --- | --- |
| 术语/缩略语 | 描述 |
| JavaScript | JavaScript（简称JS）是一种基于原型编程、解释型或即时编译型的编程语言，支持面向对象、命令式和函数式编程范式，最开始用作开发Web页面的脚本语言，后来也运用到一些非浏览器环境中。 |
| HTML | 超文本标记语言（HyperText Markup Language）的缩写，HTML文本是由结构嵌套的HTML标签组成的描述性文本，HTML标签可以说明文字、图形、动画、声音、表格、链接等，可以引用其他位置上的资源，是目前用于网页内容编写的唯一语言。 |
| Node.js | Node.js是基于Chrome V8引擎开发、跨平台的JavaScript运行环境，采用事件驱动、非阻塞式I/O的模型，可以让JavaScript运行在服务器端。 |
| node | Node-RED流程中的一个节点，它们可以在流程执行时实例化，并接收、处理和发送消息。编码上node由一组定义节点功能的JavaScript文件和用于节点配置编辑页面及帮助文本页面的HTML文件构成。 |
| flow | Node-RED中的流程，一个流程由多个node组成，node之间有数据流的依赖关系，流程的起始节点一般为事件监听相关node。流程部署后，由事件（例如手动、定时器、HTTP请求等）触发流程的执行 |
| Markdown | Markdown是一种轻量级标记语言，它允许人们使用易读易写的纯文本格式编写文档，然后转换成有效的XHTML（或者HTML）文档。 |

## 参考资料

1. Node-RED 官方网站 https://nodered.org/
2. Node-RED 官方文档 https://nodered.org/docs/
3. Node.js 官方文档 https://nodejs.org/en/
4. 钉钉开发平台官方文档 https://ding-doc.dingtalk.com/doc#/serverapi3/rg1occ
5. 飞书开放平台官方文档 https://open.feishu.cn/document/ukTMukTMukTM/uITNz4iM1MjLyUzM
6. Roger S.Pressman. 软件工程——实践者的研究方法[M]. 机械工业出版社, 2011.
7. Leszek A.Maciaszek. 需求分析与系统设计[M]. 机械工业出版社, 2009.
8. Nicholas C.Zakas. JavaScript高级程序设计[M]. 人民邮电出版社, 2012.
9. Mike Cantelon等. Node.js实战[M]. 人民邮电出版社, 2014.

## 标准、条约和规定

本项目将以下面这些标准作为参考：

1. GB/T 13702-1992 计算机软件分类与代码
2. GB/T 19003-2008 软件工程
3. GB/T 5538-1995 软件工程标准分类法
4. GB/T 9385-2008 计算机软件需求规格说明
5. GB/T 5532-2008 计算机软件测试规范
6. GB/T 18221-2000 信息技术程序设计语言
7. GB/T 8567-2006 计算机软件文档编制规范

# 项目概述

## 2.1项目目标

从Node-RED平台出发，在该平台基础上提出消息聚集和图形可视化拓展的需求并进行拓展，编写相应的需求规格说明书、测试需求规格说明书，并根据规格说明书与测试需求规格说明书对拓展内容进行模块化测试，构建相应的测试报告。

## 2.2项目可交付内容

1. 需求规格说明书。其中包含的主要需求为，基于Node-RED平台，开发消息聚集拓展包以及绘图和Markdown模板拓展包。为了支持主要需求，需求规格说明书中亦包含以下两组node拓展需求：利用钉钉和飞书的开放API的一组消息聚集node拓展；一组基于图表的数据可视化和Markdown模板渲染的node拓展
2. 满足需求规格说明书的Node-RED扩展node及基于Node-RED平台实现的消息聚集的示例应用
3. 测试需求规格说明书
4. 基于测试需求对各扩展node进行测试的测试报告

## 2.3工作内容

1. 学习使用Node-RED平台和Node.js语言
2. 学习调研需求的必要性，细致分析实现需求细节，完成需求文档
3. 基于Node-RED平台，开发项目可交付内容中的两组node拓展，并进行整合，根据实际情况，开发消息聚合的示例应用。同时根据评审和工程中出现的新需求、新问题，进行项目开发和文档编写的迭代。
4. 利用Node-RED项目的测试框架对新开发的node进行测试，完成测试报告
5. 根据课程的整体安排和每个实验的具体要求，依照项目时间计划完成其他各类文档和报告。

## 2.4项目及软件工程综合实验应交付成果

**表 2 应交付成果**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 名称 | 交付时间 | 要求 | 备注 |
| 1 | 项目计划书 | 2020.3.18 | 完整，明确 |  |
| 2 | 项目进度表 | 待定 | 完整，明确 |  |
| 3 | 需求规格说明书 | 第3，4周 |  |  |
| 4 | 需求检查单 | 第5，6周 |  |  |
| 5 | 测试需求规格说明书 | 第11周 |  |  |
| 6 | 软件测试分析报告 | 第12周 |  |  |
| 7 | 软件测试评审报告 | 第13，14周 |  |  |
| 8 | 项目工作量分析报告 | 第15，16周 |  |  |
| 9 | 项目进度分析报告 | 第15，16周 |  |  |
| 10 | 配置管理报告 | 第15，16周 |  |  |
| 11 | 项目总结报告 | 第16周 |  |  |
| 12 | 会议记录 | 第16周 |  |  |
| 13 | 项目代码 | 第16周 |  |  |

## 2.5项目验收方式和依据

验证方式：

项目验证方式采用中期课堂评测和终期交付验收。

验收证据

验收应依据“应交付成果”清单进行验收。

# 项目团队组织

## 3.1项目结构

由于项目中的不同node之间功能有相似性，按照功能可以将项目划分为三个主要部分。

数据处理node：主要完成对数据的中间处理，如对数据的统计和储存，以及绘图显示编码等操作。

消息聚集node：利用与工作软件的API接口，将消息以适当的格式聚集到工作软件中，实现最终的消息聚集。

示例应用：整合上面两组node，通过实用的场景案例展示项目所做的node拓展包的功能实用性。

## 3.2组织划分

由于项目可以按照功能划分为三个部分，为保证各个部分工作的完成质量，我们计划将组内成员也分为三个小组，分别为消息聚集组、数据可视化组以及文档和示例开发组，分别对应负责上述三个模块的开发。

## 3.3人员分工

**表 3 人员组成表**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 学号 | 姓名 | 角色 | 所属子项目组 | 主要任务 |
| SY1906416 | 暴明坤 | 组长 | 消息聚集组 | 文档编写，编码，测试，项目管理 |
| BY1906051 | 夏欣怡 | 组员 | 文档和示例开发组 | 文档编写，编码，测试 |
| SY1906113 | 胡俊涛 | 组员 | 数据可视化组 | 文档编写，编码，测试 |
| SY1906305 | 张雨濛 | 组员 | 文档和示例开发组 | 文档编写，编码，测试，会议记录 |
| SY1906118 | 叶柏威 | 组员 | 数据可视化组 | 文档编写，编码，测试 |
| BY1906028 | 刘子渊 | 组员 | 消息聚集组 | 文档编写，编码，测试 |

## 3.4协作与沟通

由于处在特殊时期，所以目前之间见面会晤的协作方式已经不太可行，协作与沟通主要以线上手段为主。

团队主要使用的线上协作工具是字节跳动公司开发的飞书，主要原因是其功能相对齐全，具有线上会议，共享文件，云文档等多种功能，同时配合微信群随时交流。同时使用Git作为项目管理工具。

团队的周期性沟通制度，确定为一周两次定期线上会议，应当在会议上明确当前进度，说明工作成果并商议计划下一阶段的工作。

a. 第一次在周二的晚上

b. 第二次在周六或周日的晚上

# 实施计划

## 4.1计划分析

1. 了解Node-RED

由于项目是基于Node-RED平台进行拓展的，所以首先应当全面了解Node-RED，熟悉其使用方法并分析其源码，尽可能的从开发思想上去了解开发者的意图。

1. 需求分析及评审

获取并说明软件需求，细化并严谨地定义指定的软件需求；采用规范的评审方法对需求进行验证和评审，并依据评审意见对需求进行修改和完善，最终完成需求规格说明书及需求检查单。

1. 测试及评审

定义测试需求并设计相关的测试用例，采用平台提供的测试框架对开发内容进行测试；采用规范的评审方法，确认测试需求、测试方法和测试结果，最终完成软件测试分析报告及软件测试评审报告。

1. 功能实现

需求功能可以分成两个部分：数据处理node和数据聚集node，由于这两部分存在一定的独立性，考虑在基于一些共同的编码规范上六个人并行作业，最后进行联调整合，并搭建场景案例验证实用性。

1. 其他软件管理工具学习

在软件研发过程中，要采用有效的方法对项目进行实时跟踪，管理项目进度及团队成员工作情况，这就需要一些专业的管理工具。因此，我们将在项目研发的同时学习相关管理工具的使用，以便能够更好地辅助项目进行，并同时监督每一位成员的工作情况。

## 4.2总体进度计划

**表 4 总体进度计划表**

|  |  |  |
| --- | --- | --- |
| **主要工作** | **周数** | **日期** |
| 进行实验准备，确定项目目标、计划及人员分工等 | 第2周 | 2020.03.06-2020.03.12 |
| 软件需求分析 | 第3-4周 | 2020.03.13-2020.03.26 |
| 软件需求评审、复审 | 第5-6周 | 2020.03.27-2020.04.09 |
| 软件项目管理、配置管理及追踪与分析 | 第7周 | 2020.04.10-2020.04.16 |
| 软件产品设计与实现 | 第8-10周 | 2020.04.17-2020.05.07 |
| 软件测试需求分析 | 第11周 | 2020.05.08-2020.05.14 |
| 软件测试评审、复审 | 第12-13周 | 2020.05.15-2020.05.28 |
| 软件测试演示及测评 | 第14周 | 2020.05.29-2020.06.04 |
| 软件项目管理、配置管理及分析与总结 | 第15-16周 | 2020.06.05-2020.06.18 |

## 4.3各小组进度计划

1. 消息聚集组
2. 对钉钉以及飞书平台的开放API进行调研 (3-4周)
3. 参考Node-RED平台现有的chatbot node，列出常用的API (5-6周)
4. 对常用API进行实现（本土化) (8周)
5. 整合到Node-RED平台 (9周)
6. 与其他组进行联调 (9-10周)
7. 测试该组node (11周)
8. 数据可视化组
9. 调研常用数据图(饼状图，折线图，散点图，柱状图等)的数据格式 (3-4周)
10. 对实现绘图和模板引擎相关的node.js基础库进行研究 (3-4周)
11. 实现常用数据图的模板引擎 (8周)
12. 整合到Node-RED平台 (9周)
13. 与其他组进行兼容测试 (9-10周)
14. 测试该组node (11周)
15. 文档和示例开发组
16. 负责监督并引导所有的文档工作 (1-16周)
17. 进度计划与控制 (1-16周)
18. 对node-red的常见用例进行调研 (3-4周)
19. 完成设计，对各节点接口的进行调研 (3-4周)
20. 完成示例的原型开发 (8周)
21. 监督其他两个组的接口兼容情况 (8周)
22. 完成示例开发 (9-10周)
23. 测试 (11周)

# 关键问题

1. 如何使用钉钉以及飞书的API, 如何在Node.js环境上进行调用。
2. 如何在Node.js环境下进行图表的绘制
3. 如何在Node.js环境下将绘制的图标输出为标准图片格式(png, jpg,...)
4. 如何实现基于Markdown的模板引擎
5. 如何将新增功能整合到Node-RED