

基于 Web 组态的通用污水处理集控 监视系统设计与应用

彭龙江^{1,2}

- (1. 上海西派埃智能化系统有限公司, 上海 200233;
2. 上海工业自动化仪表研究院有限公司, 上海 200233)

摘要: 由于大型污水处理集团存在水厂数量多、位置分散、生产工艺路线差异大, 数据刷新不及时造成辨别困难、误报警问题以及各单位之间存在着生产和管理数据的“数据孤岛”问题, 上层管理单位和人员难以对下属水厂的生产运营进行统一有效地监管、决策和调度; 为了解决上述问题, 基于 Web 的组态监控技术, 采用分层架构和模块化的设计方法设计和实现了包括配置、展示、控制以及公用模块的通用污水处理集控监视系统, 使用 MQTT 数据传输协议进行生产数据的实时更新; 通过性能测试和分析可知, 实现的系统具有较高的实时性和稳定性以及较低的资源占用率; 系统作为生产工艺和设备的集控监视子系统已经成功集成应用于上海某大型污水处理公司的生产运营管理平台以及广东某大型环保产业集团的智慧运营管理平台中; 系统运行平稳可靠, 很好地解决了存在的问题, 具有较高的推广使用价值。

关键词: Web 组态; 污水处理; 集控监视; MQTT; JMeter

Design and Application of Web-Configuration-Based Centralized Control and Monitoring System for General Wastewater Treatment

PENG Longjiang^{1,2}

- (1. Shanghai SIPAI Intelligent Systems Co., Ltd., Shanghai 200233, China;
2. Shanghai Institute of Process Automation & Instrumentation Co., Ltd., Shanghai 200233, China)

Abstract: Large-scale wastewater treatment groups have the characteristics of multiple water plants, dispersed locations, significant differences in production process routes, difficulty in distinguishing due to delayed data updates, false alarms, and data silos in production and management data between units. It is difficult for the upper management units and personnel to carry out effective supervision, decision-making and scheduling of the production and operation of subordinate water plants. To solve above issues, using web-based configuration monitoring technology and layered architecture and modular design methodology to design and implement a common wastewater treatment monitoring system including configuration, display, control and utility modules, and applying the message queuing telemetry transport (MQTT) data transmission protocol to update the production data in real-time. Through performance tests and analyses, it can be seen that the implemented system has high real-time and stability as well as low resource consumption. As a centralised control and monitoring sub-system for production processes and equipment, the system has been successfully integrated and applied in the production operation management platform of a certain large-scale sewage treatment company in Shanghai and the intelligent operation management platform of a certain large-scale environmental protection industry group in Guangdong. The system runs smoothly and reliably, solves existing problems well, and has high practical and promotion value.

Keywords: web configuration; sewage treatment; centralized monitoring; MQTT; JMeter

0 引言

组态监控 (CSC, configuration supervisory control)

技术作为工业自动化生产过程中信号数据采集监测和指令下发操作控制的关键技术手段, 具有功能配置柔性、系统集成灵活^[1]、数据更新及时^[2-3]等特点。业务操作

收稿日期: 2024-07-10; 修回日期: 2024-08-21。

作者简介: 彭龙江(1989-), 男, 硕士, 高级工程师。

引用格式: 彭龙江. 基于 Web 组态的通用污水处理集控监视系统设计与应用[J]. 计算机测量与控制, 2025, 33(8): 197-205.

人员进行简单的配置即可快速实现监控画面的二次开发,满足复杂多变的现场生产业务需求。广泛地应用于能源^[3]、化工^[4]、电力^[5]、矿业^[6]、冶金^[7]、水务处理^[8-10]等诸多领域。

文献 [1] 实现了飞机机电系统模型的实时过程监控以及监控界面的图形化显示。提高了系统协同仿真的验证效率,降低了分布式协同仿真监控系统的开发难度。文献 [5] 研发的图像组态系统为电力市场多维数据的组态展示、画面联动灵活配置,高效的界面生成效率满足了对界面需求变化的及时响应,降低了运维成本。文献 [6] 采用基于多源组态技术实现的煤矿智能化管控平台,对煤矿生产控制系统工作状态的远程监控、远程诊断以及优化调度,为企业实现了降本增效的需求和目标。

污水处理行业中大型污水处理集团普遍具有下属水厂数量众多、地理位置分布分散、生产工艺路线复杂多变等特点,各生产单位存在着生产和管理数据的“数据孤岛”问题,集团难以对下属水厂进行集中监视、分散控制、数据实时共享、统一监管、决策和调度。部署在工控上位机的组态监控系统采用客户机/服务器 (C/S, client/server) 架构,为局域网内运行的封闭系统^[9],只适用于对当前水厂局域网范围内的生产运营情况进行监控。同时,采用标签轮巡方式进行更新数据时,还存在着因数据刷新不及时造成工艺生产参数前一次采样数据未及时更新到最新值以及误报警的问题^[10],画面监测数据滞后和控制指令下发更新不及时^[11]。

为解决上述问题,以期通过统筹实现区域资源配置优化推动生产经营管理精细化和集约化,实现降低生产运营成本以及提高污水处理效率和管理水平的目标。因

此,系统采用浏览器/服务器 (B/S, browser/server) 架构^[11],使用基于 Web 的组态监控技术将生产工艺过程中的相关设备参数、工艺测量点参数进行配置、展示及控制,通过网络远程实时地对现场设备状态及生产工艺情况的异常和报警进行监控、干预、决策和处理。模块化的设计方法降低了系统功能模块之间的耦合性,有效地提升了系统功能模块集成的灵活性^[12]。

1 系统整体架构

系统整体主要由污水处理行业中的生产运营管理平台和 Web 组态集控监视系统组成。以现场的生产数据流向为主线,通过感知层采集、网络层传输、数据层存储、服务层处理以及应用层使用形成 5 层架构。其中,网络层中的消息队列遥测传输 (MQTT, message queuing telemetry transport) 协议^[13]和服务层中的 MQTT 消息服务器 EMQX 是系统实现数据频率达到秒级水平的关键技术。系统整体架构如图 1 所示。

1.1 感知层采集

感知层是数据获取的源泉。污水处理厂生产现场中的液位计、流量计、水质参数检测仪等各种生产过程仪表传感器将检测到的物理量和化学量等非电信号转换成电信号,再通过可编程逻辑控制器 (PLC, programmable logic controller) 中的输入模块获取到这些电信号,经模数转换器 (ADC, analog to digital converter) 将模拟信号转换成数字信号或者数模转换器 (DAC, digital to analog converter) 将数字信号转换成模拟信号,经由中央处理单元处理和网络传输模块让系统采集获取到现场实时的生产设备和过程工艺参数数据。同时,操作人员使用系统通过 PLC 中的输出模块执行机构控制指令

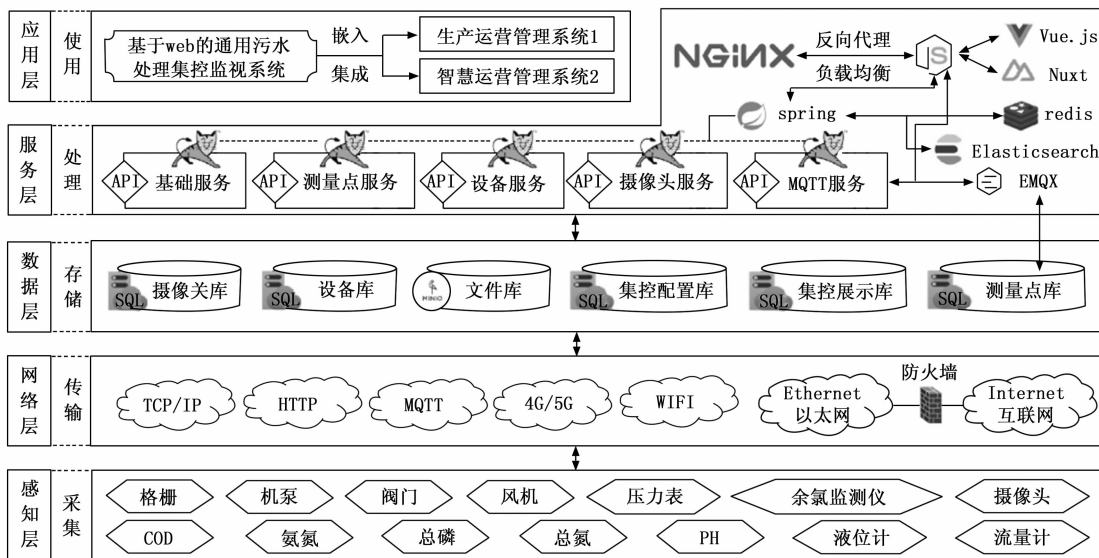


图 1 系统整体架构图

实现对生产工艺现场诸如格栅、机泵、阀门、风机等工艺单元操作执行机构的远程控制操作。

1.2 网络层传输

网络层是数据进行交互和传输的桥梁。涉及的数据传输协议主要包括 TCP/IP、HTTP、Websocket^[14]、MQTT、4G/5G、WiFi 等。系统通过使用防火墙将外部因特网与内部以太网进行网络环境的隔离，保证了用户在访问业务系统时服务和数据的安全性。

1.3 数据层存储

数据层是对数据进行存储、操作和管理的仓库。系统采用分布式存储架构^[15]，选用的数据库包括关系型数据库 SQL Server，分布式文件对象存储服务 MinIO。根据数据产生的来源设计了生产库和平台库。生产库主要负责存储生成频率高、数据量大的现场生产相关的数据；平台库主要负责存储系统维持各功能模块正常使用和运行除了生产数据的系统数据。

1.4 服务层处理

服务层是为实现各个业务模块功能而进行数据加工处理的工厂。系统采用了分布式微服务架构，前后端分离的开发模式。前端采用基于 Vue.js 的应用程序框架 Nuxt.js 以及相关技术，主要包括 JavaScript 运行时环境 Node.js^[16]、轻量级具有反向代理和负载均衡功能的高性能 Web 服务器 Nginx^[17] 等。后端采用 Java 作为开发语言以 Spring 应用框架及相关技术，主要包括不同业务功能模块处理服务、消息处理中间件、Redis 缓存、分布式搜索和数据分析引擎 Elasticsearch^[18] 以及 MQTT 消息服务器 EMQX 等。

1.5 应用层使用

应用层是对数据进行使用体现价值和产生效益的舞台。应用层中的生产运营管理系统为生产业务的正常运行和管理提供了基础平台，为生产业务功能的使用提供了人机交互界面，为集团的整体生产运营监管、分析、决策及优化提供了统一平台。

基于 Web 组态的通用污水处理集控监视系统作为一个相对独立的核心生产子系统，可灵活、快捷、方便地集成嵌入到不同污水处理生产运营管理系统当中进行使用。实现的主要功能包括：1) 多厂集控：通过组织架构树节点的切换为集团层、区域中心以及项目公司对下属业务单元的生产情况数据进行查询汇总，为管理层的集中监控、统计分析和决策工作提供基础支撑；2) 单厂监控：对单个水厂的不同生产过程工艺进行组态配置、实时监控、实时控制、异常报警、历史数据追溯等。

2 工程原型功能模块设计

基于 Web 组态的通用污水处理集控监视系统主要

由配置模块、展示模块、控制模块以及公用模块组成。工程原型的功能模块结构关系及操作流程如图 2 所示。

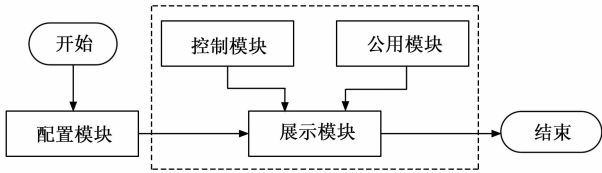


图 2 功能模块结构关系及操作流程

2.1 配置模块

配置模块主要负责配置生成每个厂区业务单元的配置数据，具有相关配置权限的人员对配置数据可读可写。主要包括基础数据的准备和维护、集控底图配置以及组态元素配置。配置操作流程如图 3 所示。

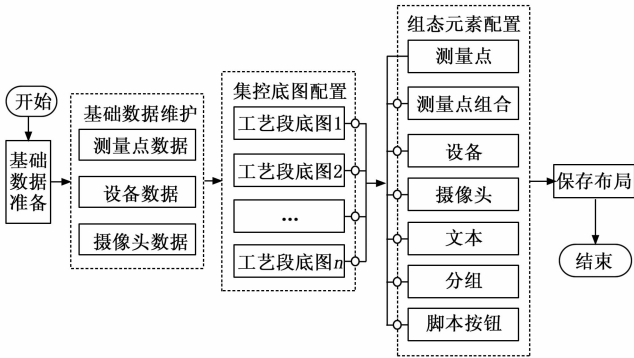


图 3 配置操作流程

2.1.1 基础数据的准备和维护

基础数据的准备和维护是为了在后续生产集控组态画面中配置组态元素时关联上生产现场中相应测量点的点位数据信息。基础数据主要包括测量点数据、设备数据、摄像头数据。数据按采集来源和途径主要分为生产现场过程仪表采集的测量点数据和固定资产类录入的设备数据和摄像头数据。

2.1.2 集控底图配置

集控底图配置主要是通过维护一份包含厂区、底图类型、标题、顺序以及图片附件信息的列表数据，其中底图类型包括有工艺详图、工艺简图以及工艺段图。结合现场生产实际需求，集控背景底图选择采用伪 3D 结构的二维平面工艺图，在大幅度降低系统资源性能消耗以及硬件设施投入成本的前提下，尽可能逼真地还原现场工艺生产线设备布置结构效果。通过设置容器元素的宽高尺寸，将工艺背景底图设置为容器元素的背景。集控底图的配置为后续的集控组态元素配置提供了含有背景画面的容器元素。

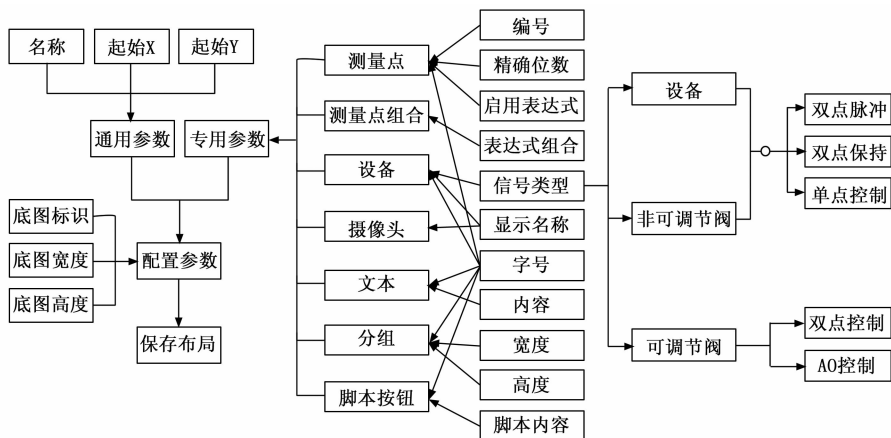


图 4 元素配置参数示意图

2.1.3 集控组态元素配置

进行集控组态元素配置时，首先需要选择到相应厂区的组织架构节点，然后选择需要配置的底图类型，设置好集控背景底图的容器元素的主要作用是各种元素资源提供一个统一调度与管理的舞台。接下来就是在设置有集控背景底图的容器元素中进行组态元素的配置。组态元素主要包括以下 7 种：分组、文字、测量点、测量点组合、设备、摄像头及控制脚本按钮。

组态元素的配置参数分为通用参数和专用参数，通用参数所有的组态元素都需要使用，专用参数只针对特定组态元素使用。单个元素配置结束保存时会绑定底图标识及底图宽高参数。其中，测量点、设备、摄像头的编号和名称是通过系统平台中相关的公用列表弹窗选择组件选择获取。通过鼠标拖拽的方式^[19]可快速方便地实现集控组态元素的位置数据（起始 X、起始 Y）更新操作。完成以上集控组态元素配置操作后，需要对整体布局进行保存更新操作。元素配置参数如图 4 所示。

2.2 展示模块

展示模块的内容和配置模块的内容基本一致，主要负责渲染展示每个厂区单元的已经配置好的数据，渲染展示的数据只读不写。主要包括元素容器、组态元素和组态设计。

2.2.1 元素容器

元素容器就是设置好集控底图的元素，负责组件元素资源的统一调度与管理。根据配置过的厂区编码及底图类型选择加载出需要展示出的监控画面。

2.2.2 组态元素

组态元素就是负责功能点输出的单元，主要包括以下元素：测量点、测量点组合、设备、摄像头、文字、脚本按钮和分组。

测量点元素展示时根据配置参数是否启用表达式进

行两种展示方式：表达式启用时显示名称数值和单位如图 5（a）所示，不启用时显示测量点图标如图 5（b）所示。

测量点组合元素展示时是根据配置的表达式组合参数进行相关测量点组合图标的显示如图 5（c）所示。

设备元素的展示会根据配置数据在话监控画面上显示设备名称，设备名称前会显示设备状态的组态图标如图 5（d）所示。

摄像头元素的展示会显示摄像头图标，同时摄像头图标上会有摄像头断线或者离线状态的颜色组态标记如图 5（e）所示。

文字元素的展示是采用黑底白字的样式如图 5（f）所示，主要负责监控画面上一些描述性内容的表达。

脚本按钮元素的展示是采用绿底白字的样式如图 5（g）所示，同时鼠标移入悬停时会显示小手样式，可进行点击操作。

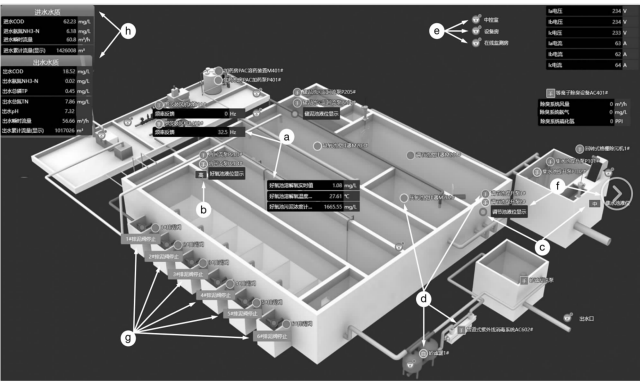
分组元素的展示主要是为了方便为其它元素集合提供一个展示背景，采用的是蓝色渐变背景元素，同时提供的分组元素名称是为这些元素集合提供的说明文字。分组元素背景处于底图里面所有元素的最下面层级如图 5（h）所示。系统集成控监视页面效果如图 5 所示。

2.2.3 组态设计

为了满足不同条件下工艺及设备的集控监视画面可灵活生成和展示渲染的需求。集控监视画面通过颜色、形状、文字以及图标等元素灵活地进行了组态设计。组态方式主要包括：颜色—形状组态、颜色—形状—文字组态、颜色—形状—图标组态以及图标组态。通过这种多维度细粒度的组态方式可以将元素的多种状态和模式集中表达呈现，很好地提升了集控组态元素状态及模式信息展示的密度。主要的元素组态如图 6 所示。

2.3 控制模块

控制模块主要是针对现场生产设备关联的测量点位



(a) 测量点不含表达式; (b) 测量点含表达式; (c) 测量点组合; (d) 设备; (e) 摄像头; (f) 文本; (g) 按钮; (h) 分组。

图 5 系统集成监视页面效果图

和生产工艺点位进行操作控制，需要通过登陆用户 id 和机器相关信息生成集控控制标识，并且完成集控控制授权才能使用。主要包括设备控制卡片和脚本控制按钮。

2.3.1 设备控制卡片

设备控制卡片主要负责对关联有测量点位的设备进行远程操作控制，远程模式下才可执行设备的远程控制操作。可通过对设备控制卡片上的开启、关闭、手动、自动元素进行相关控制模式的切换控制，设备关联点位的控制会根据信号类型执行对应的控制行为。设备的状态反馈信号包括：运行、故障、停止、就地、远程以及无信号。设备为阀门类型时，状态反馈信号包括：开到位、关到位以及无信号。控制点位的数据对象包括：模拟量输出 (AO, analog output)、模拟量输入 (AI, analog input)、数字量输出 (DO, digital output)、数字量

输入 (DI, digital input)，其中设备控制卡片界面左侧使用的 DO 点及 DI 点用于控制操作和信号状态反馈，右侧展示了 AO 点 (可修改进行下发操作) 和 AI 点 (不可操作) 的相关信息。

2.3.2 脚本控制按钮

脚本控制按钮主要负责执行一些现场生产工艺相关的操作，如紫外消毒灯运行数量的统计、一些工艺过程中阀门的启动和关闭、酸洗碱洗的触发和取消工艺流程等操作。

脚本按钮执行的原理是通过在脚本按钮元素配置时在脚本内容参数输入一条或者多条测量点位 id 的赋值语句，通过抽象语法树对脚本语句进行解析执行。需要注意的是赋值的测量点 id 变量必须属于当前厂的测量点，然后在集控画面展示状态下点击脚本按钮时按顺序执行配置好的赋值语句进行相关测量点赋值语句的执行，完成生产工艺相关的操作。

设备控制卡片及脚本控制按钮相关截图如图 7 所示。

2.4 公用模块

公用模块主要负责维持系统配置模块、展示模块及控制模块相关功能的正常运行以及对部分功能的拓展加强。公用模块主要包括：必选的关键支撑组件和可选的辅助加强组件。

2.4.1 关键支撑组件

关键支撑组件主要负责维持系统功能点的正常使用。主要包括测量点选择列表弹窗、设备选择列表弹窗和摄像头选择列表弹窗。分别用于配置模块配置集控组态元素时，测量点元素、设备元素和摄像头元素分别对

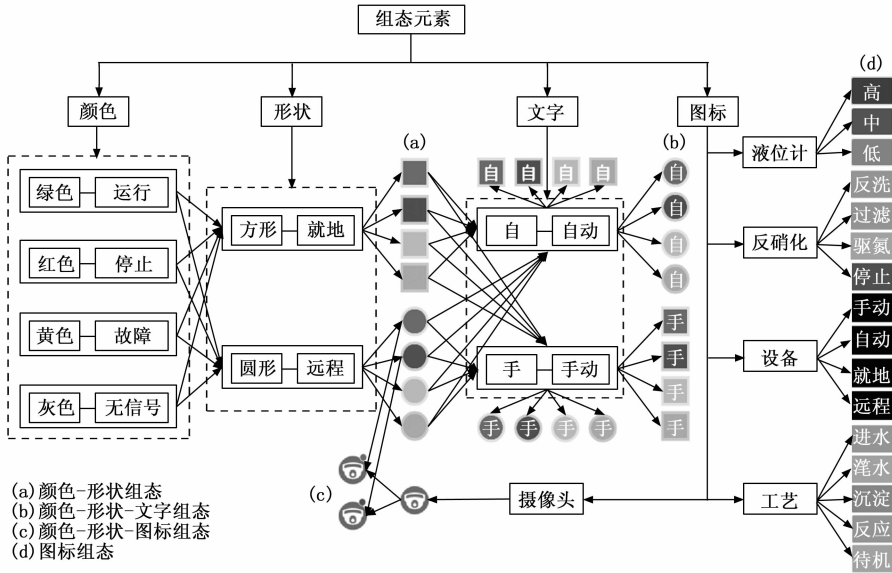


图 6 元素组态示意图



(a) 设备控制卡片相关截图



(b) 紫外消毒灯运行数量统计按钮相关截图



(c) 排泥阀一键全停按钮及点位配置相关截图



(d) 酸洗碱洗触发及取消工艺按钮截图

图 7 设备控制卡片及脚本控制按钮相关截图

应的编号和名称参数的选择获取。

2.4.2 辅助加强组件

辅助加强组件主要负责增强功能点的输出，主要包括测量点历史曲线、计算点数据透视，设备操作记录表，设备启停记录表等。

1) 测量点历史曲线查看组件。通过该组件可以快速方便地查询测量点在某一个时间范围内的历史数据曲线，方便对出现问题的测量点位数据进行追溯分析。

2) 计算点数据透视组件。通过该组件可以对计算点的汇总情况进行树状层级结构方式展示，方便对计算点位进行细粒度地分析和判断。

3) 设备操作记录组件。通过该组件可以及时准确地查阅生产设备或生产工艺的相关操作记录情况，方便定位操作人员 and 操作时间等关键信息。

4) 设备启停记录组件。通过该组件可以方便查阅现场相关设备的历史状态信息，方便查阅设备使用历史过程中的启停信息。

3 数据更新策略

合理地选择和使用数据更新策略是提升系统性能的有力手段和途径。根据实际使用过程中不同功能点数据更新的具体需求，系统中使用到的数据更新策略主要包括：HTTP 协议获取数据、setInterval 定时器轮询以及 MQTT 消息订阅。

3.1 HTTP 协议

超文本传输协议 (HTTP, hyper transfer protocol) 是一种采用客户端请求/服务器响应模式，运行于传输控制协议 (TCP, transmission control protocol) 之上进行数据传输的应用层协议。具有无状态性、支持多种数据格式、可扩展性等特点。

系统中集控监视画面初始化时通过 http 协议请求获取画面上配置过的组态元素相关的数据进行组态元素渲染，如图 8 (a) 所示。同时，设备和工艺相关测量点位的控制也是通过 http 协议进行指令下发处理的，如图 8 (b) 所示。

3.2 setInterval 定时器轮询

setInterval 定时器可以按设置好的周期 (以毫秒计) 进行数据刷新逻辑的执行，该定时器的轮询操作直到执行 clearInterval 方法 (以 setInterval 定时器方法的返回值作为参数) 时进行终止。

系统中在执行配置有多个控制点位的脚本按钮元素时使用，每个控制点位通过 HTTP 协议执行脚本语句方法的间隔周期为 2.5s，此脚本按钮配置的所有点位语句执行完毕时关闭定时器轮询，如图 8 (b) 所示和如图 8 (c) 所示。

3.3 MQTT 消息订阅

MQTT 是一种基于 C/S 架构采用主题发布/订阅模式的消息传输协议^[20]。适用于在低带宽、网络不可靠的环境中工作的远程传感器和控制设备之间进行消息的传输。

系统在初始化时会创建 mqtt 连接实例，根据厂区的网关和通讯点位进行相关主题的订阅，在集控监视画面上所有的组态元素完全渲染之后，通过点位 id 让组态元素与 mqtt 监听到的实时数据进行匹配和点位元素绑定操作，当实时数据中的点位 id 匹配到了组态元素中的点位 id 时，数据就会实时更新，如图 8 (d) 所示。

mqtt 发布时采用的是逢变则报的数据更新推送机制。

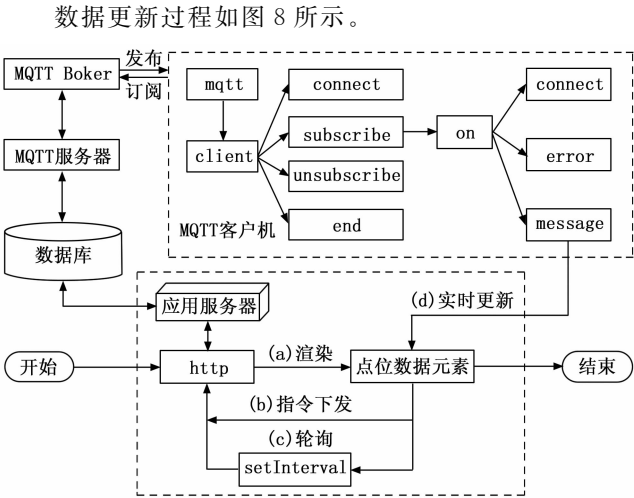


图 8 数据更新过程示意图

4 实验结果与分析

为了验证基于 Web 组态的通用污水处理集控监视系统的有效性、实时性和稳定性。对系统进行了监控功能测试以及系统性能测试。

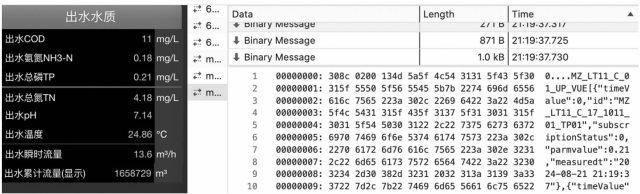
4.1 监控功能测试

通过以下操作对系统进行监控功能测试。登陆账户进入系统，选择组织树下面的厂节点，选择集控监视菜单进入单厂监视画面（默认为全厂工艺流程）。页面通过 HTTP 接口获取到监控画面的元素配置数据（主要包括测量点、测量点组合、设备、摄像头、分组元素、文字、脚本按钮）进行相关元素的渲染，同时，完成 MQTT 的创建和主题的订阅，页面接收到的数据结构主要包含了点位 id、点位值和获取值的时间，通过点位 id 与渲染在画面中的测量点位元素 id 进行匹配绑定实现数据的实时更新显示。通过 chrome 浏览器的开发者工具可实时查看点位数据的传输情况，可见数据的刷新频率达到秒级。测量点位数据实时更新测试如图 9（a）所示。设备远程控制时，需要将设备设定为远程模式，通过 HTTP 接口调用相关的操作指令（如启停、手/自动、部分 AO 点位参数的下发更改），设备执行完相关动作后，MQTT 会将实时设备状态数据反馈推送到页面上进行元素的匹配和数据的更新。部分设备的远程启停控制记录如图 9（b）所示。

经过上述监控功能测试，说明系统很好地完成了对现场生产工艺参数的监视以及对现场生产设备的远程操作控制的生产使用需求。

4.2 系统性能测试

系统性能效果的优劣主要取决于数据更新的稳定性和实时性，同时还需兼顾系统运行过程中的资源占用率



(a) 测量点位数据实时更新测试



(b) 部分设备的远程启停控制记录

图 9 系统监控功能测试图

保持在一个合理的范围内。为了验证选用数据更新策略的系统性能效果，对使用 HTTP 协议获取数据结合 setInterval 定时器轮询的数据更新策略与使用 MQTT 消息订阅模式的数据更新策略采用开源测试工具 Apache JMeter^[21]从实时性、稳定性和资源占用率 3 个方面进行性能测试，并对结果进行分析。

4.2.1 测试条件及方法

1) 硬件环境：处理器为 Intel Xeon Processor (Skylake, IBRS) 2.29 GHz；已安装的内存为 32.0 GB；系统类型为 64 位操作系统，基于 x64 的处理器。

2) 软件环境：操作系统版本为 Windows Server 2016 Standard，版本号为 22H2（内部版本号为 22621.3155）；Google Chrome 浏览器（版本 120.0.6261.112,64 位）；

3) 测试工具：Apache JMeter 5.3；MQTT XMeter 2.0.2；PerfMon Metrics Collector 2.1；

4) 测试方法：分别对使用 HTTP 协议和 MQTT 协议更新数据的策略进行性能测试。根据实际工程项目中系统实际用户使用的需求。从 10 个并发用户数开始，按 30、25、20、15 个用户数的大小间隔依次递增到最大的 100 个并发用户数。需要满足的系统性能指标要求如表 1 所示。

表 1 系统性能指标要求

性能指标	用户数	响应时间	成功率	CPU 占用率	内存占用率
要求	100 个	≤3 s	≥99 %	≤60 %	≤80 %

4.2.2 测试步骤

系统性能测试步骤主要由以下 4 步完成：1) 制定

测试计划；2) 对测试工具进行基础的参数配置；3) 根据测试计划完成相关测试项目的参数设置及结果的监听器绑定；4) 启动执行测试操作并获取测试结果。

系统性能测试计划如表 2 所示。

表 2 系统性能测试计划

计划序号	用户数/个	Ramp-Up Period/s	持续时间/s
1	10	3	5
2	40	3	5
3	65	3	5
4	85	3	5
5	100	3	5

系统性能测试过程如图 10 所示。

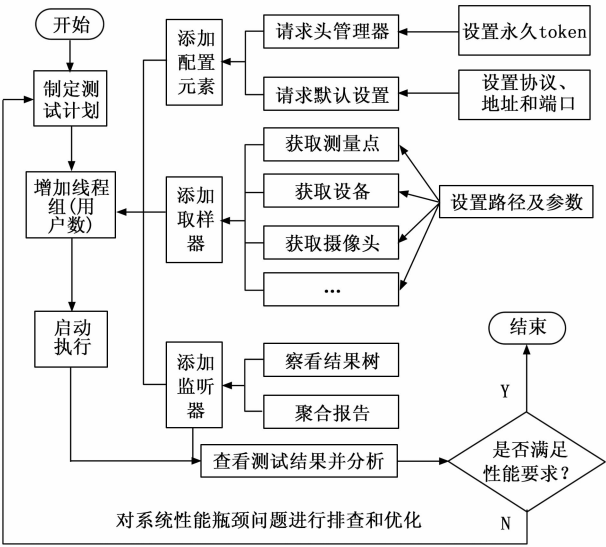


图 10 系统性能测试过程示意图

4.2.3 测试结果分析

系统性能测试结果如表 3 所示。

表 3 系统性能测试结果

性能指标		实时性			稳定性		资源占用率	
更新策略	用户数/个	响应时间/ms			成功率/%		CPU/%	内存/%
		最快	最慢	平均				
HTTP	10	39	528	124	100		32	74
	40	38	419	121	100		33	75
	65	38	661	143	100		35	76
	85	36	709	181	100		39	77
	100	37	1010	223	100		42	77
MQTT	10	99	117	106	100		29	73
	40	99	155	109	100		30	73
	65	98	126	107	100		32	74
	85	94	118	104	100		33	76
	100	97	118	105	100		35	76

根据表 3 中测试结果分析可知：HTTP 协议的最快响应时间是 36 ms，最慢响应时间 1 010 ms，平均响应时间是 158.4 ms，成功率 100%；CPU 平均占用率为 36.2%，内存平均占用率为 75.8%；MQTT 协议的最快响应时间是 94 ms，最慢响应时间是 155 ms，平均响应时间是 106.2 ms，成功率 100%；CPU 平均占用率为 31.8%，内存平均占用率为 74.4%。由此可得出以下结论：

1) 通过 HTTP 协议和 MQTT 协议这两种数据获取更新的策略都可满足基本的系统性能指标要求。

2) 通过 MQTT 协议进行数据更新的策略相较于 HTTP 协议具有更好的实时性和稳定性，以及较少的资源占用率。

3) 通过 HTTP 协议结合使用 setInterval 定时器轮询时，如果在较长时间未进行定时器销毁的情况下，系统资源占用率会出现逐渐上升的趋势^[19]。并且，这种方式的数据更新频率还依赖数据库中数据的更新频率，由于 HTTP 请求进行轮询时与数据库中数据的更新时机存在着不一致的情况，监控页面会出现部分数据更新滞后的问题。而 MQTT 协议这种方式，由于 MQTT 消息服务器 EMQX 采用的是逢变则报的数据推送机制，很好地保证了数据的实时更新频率，消除了监控页面部分数据可能出现更新滞后造成不一致的问题。

因此，在监视页面需要长时间打开的使用场景下采用 MQTT 协议。但在较短时间内进行轮询操作后销毁定时器的使用场景，对系统资源占用率产生的影响较小。因此，在进行脚本按钮操作多点位脚本语句需要按点位顺序执行时，使用 HTTP 协议结合 setInterval 定时器轮询的数据下发更新策略是可行的。

5 工程应用案例

本系统已经成功集成应用于上海市某大型污水处理公司的生产运营管理平台以及广东省某大型环保产业集团的智慧运营管理平台。前期阶段应用在上海某大型污水处理公司的生产运营管理平台时，主要集成了配置、展示和部分相关的公用模块，展示模块页面的数据更新策略从使用 HTTP 协议结合 setInterval 定时器轮询逐渐过渡到采用 MQTT 协议，经过实施使用基本达到了预期的目标和效果，明显改善了生产数据的更新频率，有效提升了业主方的生产运营管理水平，降低了运营管理成本。后期阶段在之前工作的基础上，在广东省某大型环保集团的智慧运营管理平台的建设过程中，除了集成了原有成熟的配置和展示模块，还增加了控制模块，并且对相关的公共模块进行了持续集成和优化，如设备操作记录、启停记录、测

量点历史曲线以及监控画面测量点位数据超限变色,可为生产过程中的设备和测量点出现的问题进行更好的追踪、处理、分析和预防。

系统在近几年的运行过程中,效果良好。同时在保证生产稳定运行的前提下对系统功能持续进行着优化完善。与此同时,对系统整体的性能也进行着持续优化,数据更新频率得到明显提升,达到了秒级的效果。对于现场生产设备及生产处理工艺的高效健康稳定运行起到了关键的保障支撑作用^[22-23]。

6 结束语

针对污水处理行业生产业务现状和问题,研究了组态监控系统 Web 技术实现的方式,设计和实现了一套基于 Web 组态的通用污水处理集控监视系统。数据实时更新频率提升至秒级,解决了系统数据更新不及时及误报警问题,有效地帮助上层管理单位和人员实时把控整体生产运营状况,进而进行高效决策,统筹优化调度区域资源,提高了集团整体的生产运营效率和管理水平,产生了较好的经济效益和社会效益。为相关企业契合国家数字化转型战略的业务场景进一步实施落地起到了很好的示范引领作用。

参考文献:

- [1] 樊智勇,张同,刘涛. 基于 DDS 的飞机机电系统组态监控技术研究 [J]. 计算机测量与控制, 2021, 29 (8): 104-108.
- [2] 李俊,惠力,聂睿. 试飞遥测实时监控分析报告自动生成系统设计 [J]. 科学技术与工程, 2023, 23 (32): 13889-13895.
- [3] 金美玉,汤亚玲,张学锋. 煤气泄漏无线监测预警系统及源点定位仿真研究 [J]. 计算机应用与软件, 2022, 39 (4): 95-99.
- [4] 查俊,吕运容,朱建新,等. 基于物联网的石化工业监控预警与安全应急系统 [J]. 现代化工, 2016, 36 (7): 178-180.
- [5] 颜自坚,孙鹏,刘晓梅,等. 电力市场技术支持系统人机图形组态架构及关键技术 [J]. 电力系统自动化, 2021, 45 (5): 122-128.
- [6] 张亮. 基于多源组态技术的恒姑煤矿智能化管控平台设计与实现 [J]. 煤矿安全, 2023, 54 (9): 227-232.
- [7] 李志慧,杨凌志,胡航,等. 新钢基于多元数据采集的电弧炉炼钢过程能量监控系统 [J]. 中国冶金, 2022, 32 (11): 97-105.
- [8] 邓小龙,徐大江,芦俊,等. 基于物联网的污染源监控预警系统 [J]. 给水排水, 2020, 56 (9): 123-126.
- [9] 张延顺,景加荣,宁远涛,等. 基于微信、SAE 与组态王

的移动端监控系统设计与实现 [J]. 工业仪表与自动化装置, 2020 (3): 39-42.

- [10] 周晓辉,张茂华,赵均,等. 武汉市二次供水远控平台的构架和应用 [J]. 中国给水排水, 2021, 37 (20): 41-46.
- [11] 冷波. 基于 B/S Web 架构组态监控平台的应用 [J]. 自动化应用, 2019 (9): 72-74.
- [12] JHA P C, BALI V, NARULA S, et al. Optimal component selection based on cohesion & coupling for component based software system under build-or-buy scheme [J]. Journal of Computational Science, 2014, 5 (2): 233-242.
- [13] 曹小华,李泊恒,徐上尉. 基于 MQTT 协议的物联网岸电监控系统 [J]. 计算机应用与软件, 2023, 40 (3): 11-16.
- [14] SOEWITO B, CHRISTIAN, GUNAWAN F E, et al. Websocket to support real time smart home applications [J]. Procedia Computer Science, 2019, 157: 560-566.
- [15] WANG Y J, LIU J J, HE X, et al. Design and realization of rock salt gas storage database management system based on SQL Server [J]. Petroleum, 2018, 4 (4): 7.
- [16] BERMUDEZ O J, BESADA P E, LOPEZ O J A, et al. Remote web-based control laboratory for mobile devices based on EJS, raspberry Pi and node.js [J]. IFAC-PapersOnLine, 2015, 48 (29): 158-163.
- [17] LEI Z C, ZHOU H, YES S W, et al. Cost-effective server-side re-deployment for web-based online laboratories using nginx reverse proxy [J]. IFAC-PapersOnLine, 2020, 53 (2): 17204-17209.
- [18] DHULAVVAGOL P M, BHAJANTRI V H, TOTAD S G. Performance analysis of distributed processing system using shard selection techniques on elasticsearch [J]. Procedia Computer Science, 2020, 167: 1626-1635.
- [19] 彭龙江. 基于 Web 技术的通用巡检路径绘制方法研究与应用 [J]. 制造业自动化, 2024, 46 (1): 97-104.
- [20] NURUDIN, SUKMANDHANI A A, ZARLIS M. Monitoring applications for vehicle based on internet of things (IoT) using the MQTT protocol [J]. Procedia Computer Science, 2023, 227: 73-82.
- [21] 余青. 利用 Apache Jmeter 进行 Web 性能测试的研究 [J]. 智能计算机与应用, 2012, 2 (2): 55-57.
- [22] 石启亮,李强,曾俊康. 船载统一载波测控系统基带实时可视化综合监视系统软件设计 [J]. 计算机测量与控制, 2020, 28 (4): 7-10.
- [23] 王海东,孙淑光,匡杉. 基于多假设解分离算法的 BDS/SINS 紧组合完好性监视 [J]. 计算机测量与控制, 2019, 27 (12): 249-253.