

等离子体控制系统用户界面与参数管理^①



谢文静^{1,2}, 张睿瑞¹, 肖炳甲¹, 袁旗平¹, 徐根^{1,2}, 严玲玲¹

¹(中国科学院合肥物质科学研究院等离子体物理研究所, 合肥 230031)

²(中国科学技术大学, 合肥 230026)

通信作者: 张睿瑞, E-mail: rrzhang@ipp.ac.cn

摘要: 等离子体控制系统是聚变实验中重要的系统之一, 用于对等离子体的各项参数进行实时的反馈控制。目前自主研发的等离子体控制系统基于组件模型开发, 其核心控制功能通过算法组件执行各种等离子体控制算法来实现, 需要提供图形界面编辑算法的预设参数, 并对参数进行管理, 因此基于 PyQt5 实现图形界面, 并开发参数配置组件负责参数存储回调, 通过发布/订阅通讯机制完成两者的数据传输。采用可扩展标记语言定义控制算法预设参数信息使得用户界面与控制算法解耦, 通过接口描述语言统一定义参数数据结构, 设计 MySQL 数据库存储历史参数数据, 并基于发布/订阅通讯机制完成数据通讯。

关键词: 用户界面; 可扩展标记语言; 接口描述语言; 发布/订阅

引用格式: 谢文静, 张睿瑞, 肖炳甲, 袁旗平, 徐根, 严玲玲. 等离子体控制系统用户界面与参数管理. 计算机系统应用, 2023, 32(10): 115-122. <http://www.c-s-a.org.cn/1003-3254/9242.html>

User Interface and Parameter Management of Plasma Control System

XIE Wen-Jing^{1,2}, ZHANG Rui-Rui¹, XIAO Bing-Jia¹, YUAN Qi-Ping¹, XU Gen^{1,2}, YAN Ling-Ling¹

¹(Institute of Plasma Physics, Hefei Institutes of Physical Science, Chinese Academy of Sciences, Hefei 230031, China)

²(University of Science and Technology of China, Hefei 230026, China)

Abstract: Plasma control system is one of the critical systems in fusion experiments, responsible for real-time feedback control of various plasma parameters. The existing independently developed plasma control system adopts a component-based model, with its core control function implemented by executing various plasma control algorithms through algorithm components. It requires preset parameters of the graphical interface editing algorithm and will manage them. Therefore, the graphical interface is implemented by using PyQt5, and a parameter configuration component is developed to process parameter storage and retrieval. Data transmission between them is achieved through publish/subscribe messaging mechanism. Extensible markup language is used to define the preset algorithm parameter information, so as to decouple the user interface and control algorithm. The parameter data structure is uniformly defined through interface description language, and MySQL database is designed to store historical parameter data. Data communication is completed based on publish/subscribe messaging mechanism.

Key words: user interface; extensible markup language; interface description language; publish/subscribe

托卡马克装置是目前最有希望实现受控核聚变从而解决能源危机的实验装置, 其物理实验的开展依赖于高效稳定的等离子体控制系统 (plasma control system,

PCS)^[1]. 正在设计的下一代 PCS 是面向中国聚变工程实验堆 (China fusion engineering test reactor, CFETR) 开发的^[2], 基于组件架构设计, 对控制系统的核心部分

① 基金项目: 国家磁约束核聚变能发展研究专项 (2018YFE0302100); 国家自然科学基金 (12075285, U1867222)

收稿时间: 2023-01-29; 修改时间: 2023-04-10; 采用时间: 2023-04-23; csa 在线出版时间: 2023-07-14

CNKI 网络首发时间: 2023-07-17

进行抽象和分离,形成等离子体控制框架.控制框架有丰富的应用组件,如 workflow 组件用于协调各个组件之间的工作流程,参数配置组件用于管理 PCS 的预设参数,日志组件用于记录系统及实验日志等.同时,框架提供任务分配、进程同步、数据通讯等功能,根据装置运行需求,需要依托等离子控制框架发展更多的控制算法来实现 PCS 对等离子体的控制.

随着诊断手段和激励系统的增加,更多的控制算法会被集成到控制系统中,因此需要开发用户界面作为系统前端,实现算法预设参数的有效输入,并开发参数配置组件负责预设参数的高效管理.

1 研究现状与需求分析

1.1 研究现状

韩国 KSTAR 装置针对常用操作,开发了 KWT 控件库以实现 EPICS 数据系统的不同访问通道^[3],开发人员基于 KWT 复用编程,可灵活实现不同的操作界面,对实验数据进行处理与查看.

华中科技大学主研的 J-TEXT 装置基于模型-视图-控制器 (model-view-controller, MVC) 模式开发模块化框架^[4],将与 EPICS 的数据交互操作定义为模型,在此基础上复用功能相似的视图类,开发程序界面时可复用功能相似的视图类,并要求尽可能继承或实现特定的类或接口,最大程度地避免冗余和独立的代码.

中科院等离子体物理研究所研制的 EAST 装置,其等离子体控制系统继承发展自美国的 DIII-D 装置等离子体控制系统^[5],主要模块通过 C 语言编写,用户界面使用交互式数据语言 (interactive data language) 实现.当控制算法更新时,既需要修改 C 语言文件,也可能需要 IDL 编写其独立界面框,对算法开发人员来说学习门槛较高,使用体验不佳.

1.2 需求分析

PCS 用户界面与参数管理的功能性需求主要包括

以下 3 个部分.

(1) 用户界面需要与具体算法解耦.用户使用用户界面选择参与当前实验的控制算法及设置算法具体的预设参数.用户界面的布局和生成方式不应随着算法的更新而改变.因此界面的设计需要与具体算法解耦,使得算法开发人员专注于具体的算法逻辑实现,而无需担心界面生成的方便与否.

(2) 多用户模式.控制系统在实验时会有多个终端同时设置数据的情况,需要考虑界面与等离子体框架之间有良好的通信机制,能够及时将用户的修改通知其他用户,并且快速同步数据.

(3) 存储并回调功能.每次实验中控制系统需要设置上千条参数,逐一设置会给操作人员带来很大的工作量.因此需要有设置历史数据存储及回调功能,在用户界面上以历史设置为基础进行修改,形成下一次实验的参数预设.

1.3 技术选型

Qt 是一个易用高效的图形用户界面 (graphical user interface, GUI) 框架,通过 Qt Designer 可以设计 GUI 程序界面,大大提高开发效率^[6].控制算法开发人员大多是物理研究人员,了解和使用 Python 语言.基于此我们采用了 Python 和 PyQt5 设计开发了控制系统的用户界面,使用接口描述语言 (interface description language, IDL) 定义统一的数据接口,基于 ZeroMQ 开发订阅/发布的消息通讯模式,采用 MySQL 数据库存储预设参数,并通过参数配置组件统一管理参数的存取与传输.

整体设计如图 1,主要包括用户界面 (user interface, UI)、参数配置组件 (configuration server, CS) 以及数据库 (database, DB) 这 3 部分功能.在实验准备阶段,用户通过 UI 与 CS 通信实现对算法预设参数 (algorithm parameters) 的编辑,实验开始时通过 CS 将当前所有算法预设参数写入数据库进行存储,并发往实时 (realtime, RT) 组件,供实时工作结点进行算法执行.

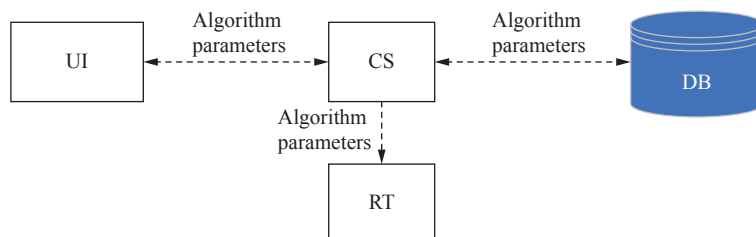


图 1 系统架构

2 界面设计与实现

2.1 控制算法介绍

PCS 按照被控对象的不同, 将等离子体电流、位形、密度等控制内容划分为不同的控制目录 (category). 每个控制目录有多个控制算法 (algorithm), 在实验中被用于不同放电阶段 (phase), 因此一个 phase 包含了指定的控制算法以及该算法执行过程中各时刻所需的数据^[7]. 并且同一个控制算法在整个放电过程中可能会被多次使用, 需要设置控制算法参与的执行序列, 即放电时序 (sequence).

用户界面中需要组织、呈现的算法结构大致如图 2 所示, 每个控制目录下包含多个放电时序, 放电时序则由多个放电阶段构成. 并且由于控制算法的数据项较多, 根据其间的关联性划分为不同的子集 (subset), 子集内的预设参数则具体分为两类: 一类是随时间变化的波形数据, 由一组顶点组成; 一类是不随时间变化的静态数据, 如运行模式、采集信号标定系数设置等.

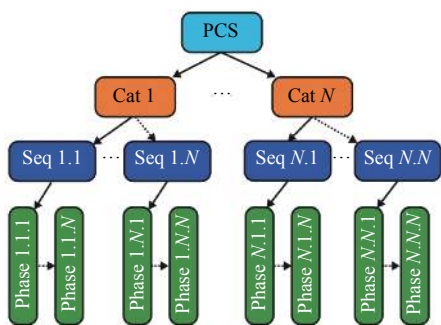


图 2 PCS 控制算法结构

2.2 算法呈现与解析

可扩展标记语言 (extensible markup language, XML), 利用自我描述的方式定义数据结构, 描述信息包括各类数据及数据间的关联关系. 选择 XML 配置控制算法, 因为其能够有效解决类别繁多、控制过程复杂的问题^[8], 并且其结构性可以显著降低算法人员开发算法的门槛. 此外, XML 的自我描述性使得其可以作为控制系统的数据库之一, 同时又作为版本文件供存档查阅, 保持完整的一致性, 其作为模型和数据存储的文件, 已经得到了广泛的应用^[9-13].

图 3 所示, 是一个控制算法的定义片段. 首先是算法的描述信息, 包括名称、功能描述、别名、算法数

据类名, 其次是子集划分信息, 子集内包含一系列波形 (waveform) 以及静态数据 (staticdata). 此处为节省篇幅, 只列出一段波形定义. 每个波形统一由描述字段、值范围、显示范围、默认值组成.

```

<?xml version="1.0" encoding="utf-8"?>
<algorithm>
  <name>coilcurrent</name>
  <descr>Coil Current Control</descr>
  <algalias>cc</algalias>
  <algclass>ipp::alg::cc::coilcurrent</algclass>
  <subsets>
    <subset>
      <name>Targets</name>
      <waveform>
        <name>PF1</name>
        <varname>cc.t.pf1</varname>
        <xlabel>time (seconds)</xlabel>
        <ylabel>kA</ylabel>
        <descr>PF1 coil current</descr>
        <step>0</step>
      </scale>
      <xmin>-1.1</xmin>
      <xmax>10.1</xmax>
      <ymin>-15.0</ymin>
      <ymax>15.0</ymax>
    </scale>
    <yrange>
      <ymin>-14.5</ymin>
      <ymax>14.5</ymax>
    </yrange>
    <vertices>
      <vertex>
        <x>0</x>
        <y>0</y>
      </vertex>
    </vertices>
  </waveform>
  ...

```

图 3 波形 XML 示例

静态数据则相对繁琐, 其包含的数据类型不统一, 界面呈现方式也不完全一致. 综合考虑后, 将静态数据分为两大类, 一类提供通用的模板化开发方式, 一类提供自定义的开发方式.

针对现有的需求, 开发了 3 类模板, 分别为 Parameter、Filter、Matrix. 以 Parameter 为例, 图 4 是一段 Parameter 数据项的 XML 示例, 表 1 详细说明了其编写规则, 算法开发人员依据实际需求设计数据项中各项属性的类型、默认值等, 界面通过通用程序来读取, 即可解析数据生成弹出窗口.

以图 5 的窗口为例, XML 中 staticdata 的“name”字段值作为基础设置窗口标题, 然后以“item”中的内容添加参数控件. 一个参数控件由一个编辑控件以及一个标签控件组成. XML 中 type 为“int”或“float”时, 添加

文本编辑框;为“enum”,使用下拉框来选择项;为“bool”,则使用复选框来标识是否勾选^[14].右侧的标签控件用来标识该项所编辑的数据含义.“rowlayout”字段是指示界面每行展示的参数控件个数,图5中每行仅有一个参数控件.参数控件下方是每个弹出窗口共同的按钮,控制数据的发送、加载、更新等功能.

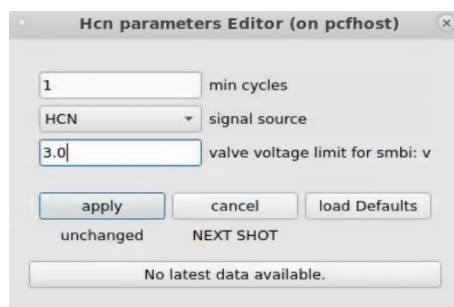


图5 Parameter 界面开发示例

```

<staticdata>
<name>Hcn parameters...</name>
<varname>df.b.Hcn</varname>
<step>1</step>
<rowlayout>1</rowlayout>
<descr>Hcn</descr>
<item>
<name>mincycle</name>
<type>int</type>
<label>min cycles</label>
<default>2</default>
<values>2</values>
</item>
<item>
<name>sigsrc</name>
<type>enum</type>
<label>signal source</label>
<default>HCN</default>
<values>HCN|FIR</values>
</item>
<item>
<name>volforsm</name>
<type>float</type>
<label>valve voltage limit for smbi: v</label>
<default>4.0</default>
<values>4.0</values>
</item>
</staticdata>
    
```

数据名称
变量名
算法类别
布局
描述信息

子项描述信息:
数据类型
界面标签
默认值
实际值

图4 Parameter XML 示例

表1 Parameter XML 描述

XML节点	XML标签	说明
Parameter	name	算法数据项名称,作为弹出窗口的标题
	step	标识staticdata类型, 1-Parameter, 2-Filter, 3-Matrix, 4-自定义editor
	descr	该算法数据项的功能说明
	rowlayout	界面每行展示的item个数
	item	每个子数据的具体信息
item	name	该项名称
	type	子数据项数据类型,依据数据类型选择不同的界面控件呈现
	label	子数据项描述内容,界面中以标签控件呈现
	default	子数据项数据值
	value	子数据项数据可选值,枚举类型时使用

对于无法通过模板开发的静态数据,提供另一套方法.算法设计人员可以使用 PyQt 提供的 Qt Designer,通过拖拽控件的方式绘制合适的操作界面,并生成 Python 文件.这些自定义的界面文件统一存放在自定义包 Editor 中,并在包目录的“__init__.py”中引入界面模块,随后只需在 XML 中记录该文件中界面模块的 class name 即可.

执行过程如代码 1 所示,主界面读取 XML 时,根据 step 的值判断为自定义界面,便利用 Python 的动态引入特性^[15],实时引入 Editor 包.之后从 XML 中读取自定义编辑器的类名字段值,依据类名,对 Editor 调用对象反射函数^[16],得到窗口的实例化并进行展示.便可以完成根据类名字符串得到该类实例的模板化调用过程,有效集成自定义的静态数据.

代码 1. 自定义界面生成方式

```

if Customized Interface then:
import Editor
read XML to get Classname
Window ← Instance of Customized Interface
show Window
    
```

采用 XML 方式有利于控制算法版本控制,结构稳定,易于修改,但是算法开发人员手工编写 XML 易错不直观.因此专门开发了软件开发平台设计算法^[17],通过可视化拖拽、仅填充具体数据的方式来自动生成 XML 和向控制框架的数据引擎注册变量.图6为软件开发平台开发 Parameter 类静态数据项的示例,只需依次填入名称、描述、布局的值,并依据需求添加 item 信息即可,有效简化了开发过程,提高了算法的开发效率.

2.3 界面实现

用户界面通过加载所有算法的 XML 文件获取

每个预设数据项的描述定义,并分别按照“category-phase-subset-dataitem”以及“category-sequence”的结构将数据项展示在界面上,此处选用 TreeWidget 作为界面容器.

如图 7 所示,界面启动后,得到树形算法结构.若当前选中的底层数据项以“...”结尾,则说明该项为静态数据,用户点击后,界面根据其所属类型生成对应的弹窗供用户编辑.

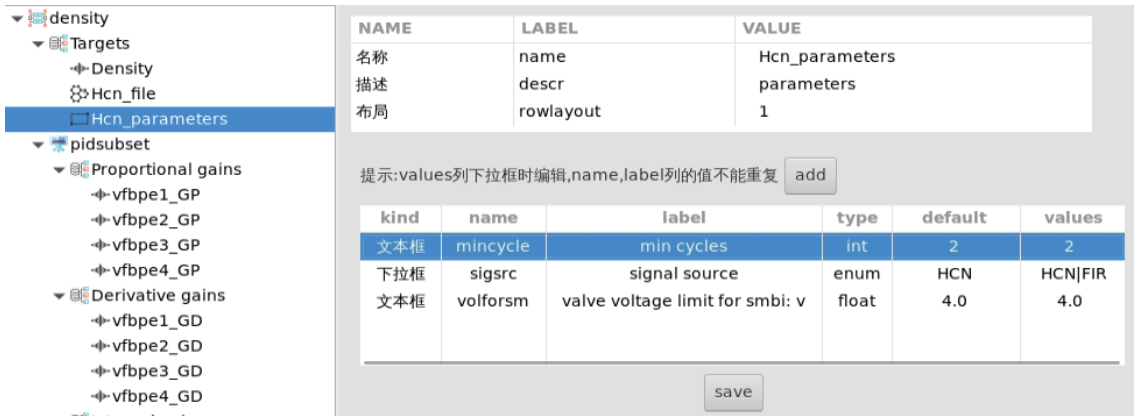


图 6 软件开发平台可视化开发 Parameter 类静态数据项示例

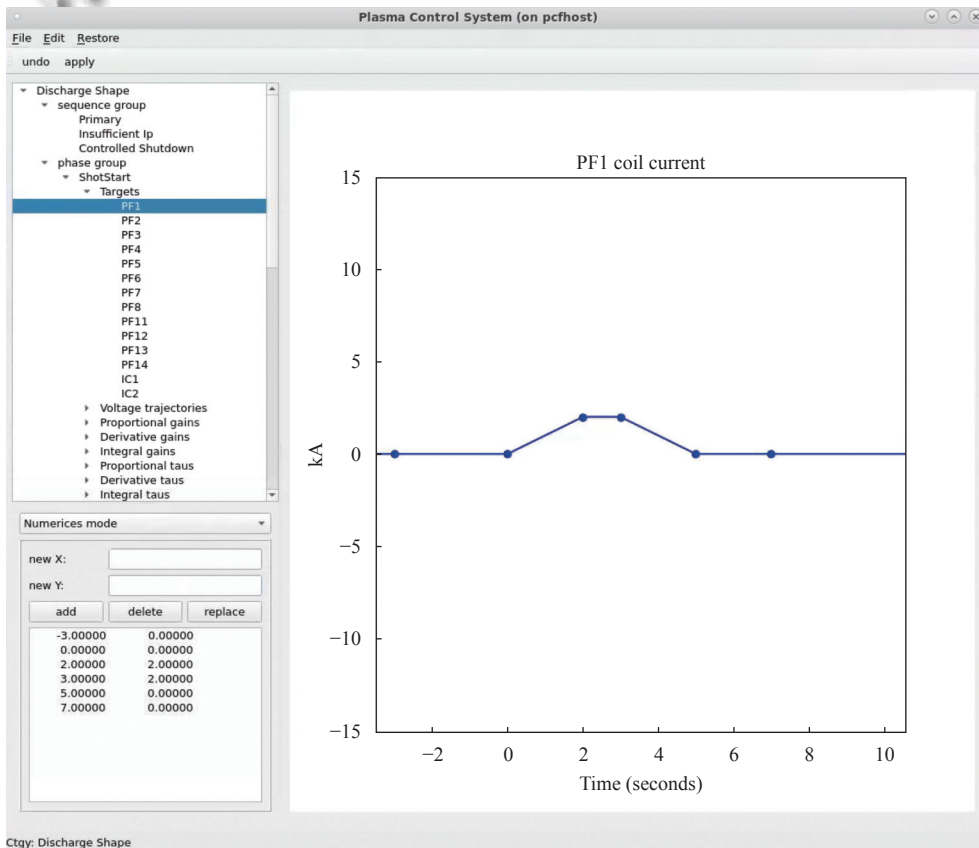


图 7 PCS 操作主界面

若当前项不包含“...”,说明选中的是波形数据,主界面右侧是根据其存储的顶点值绘制折线图,顶点之

间的数据值按照线性插值方式计算得到.算法树下方是顶点的编辑区,可以新增、删除并替换顶点.

3 数据通讯与管理

操作人员在用户界面设置的控制算法参数应作为输入数据,传输至等离子体控制框架,经过算法组件的执行,得到控制结果,并且所有设置的数据需要与放电实验炮号关联,存储至数据库备用。基于等离子体控制框架的设计模式,开发参数配置组件,与界面进行通讯,接收、分发预设参数,并存取数据库。当所有参数设置完毕后,还需要将数据发送至实时调度组件,供其执行使用。由于涉及多方数据传输,且需要跨编程语言通讯,并且所需数据格式并不完全一致,因此通过IDL定义各个接口,自动生成接口代码,在ZeroMQ的基础之上开发订阅/发布通讯模式,降低系统通讯负担,提高通讯效率。

3.1 数据接口定义

等离子体控制系统框架采用IDL定义和规范整个内部和外部的接口关系,包括基本接口数据类型生成,订阅/发布接口代码生成,以及序列化/反序列化代码生成等。IDL属于对象管理组织提出的CORBA(common object request broker architecture)规范,用于声明对象中的接口。它提供了一套通用的规范数据类型,通过组合、嵌套实现复杂数据类型的定义,再映射为所需的编程语言,实现与平台无关的顶层设计开发^[18]。用户界面设置的数据需要传递给参数配置组件。系统以IDL语言为基准,定义原始数据结构,通过等离子体控制系统框架的IDL编译器,将IDL映射为C++以及Python语言的代码,既可以省去人工编写数据类的时间,也可以降低出错概率。

详细来说,编译器生成的数据类都提供成员的访问器和修改器。特殊的,C++数据类中,提供了数据对象序列化为JSON数据流以及从JSON数据流反序列化为数据对象的两种方法。Python版本的数据类提供以字典作为参数的构造函数,若该字典不为空,则用其来更新类的“__dict__”属性,便可修改字典中包含的类成员为目标值。进一步的,在Python中,JSON数据流是以键值对字符串的形式存储的,因此Python数据类的序列化与反序列通过“__dict__”属性与JSON数据流的相互转化来实现。

3.2 消息订阅/发布

目前EAST PCS在消息传输中使用的是轮询方式,界面每隔一段时间便向PCS的波形服务进程查询是否有数据更新,若有则获取最新的数据,使得系统通讯的

实时性依赖于轮询的间隔,并且无效轮询会消耗不必要的资源。因此本系统改用订阅/发布(pub/sub)方式来处理消息通讯,仅当消息队列有消息时,才会推送该内容,实时性更高,并且不消耗额外资源。

传统pub/sub模式适用于单个发布者与多个订阅者的场景,发布者与其所有的订阅者都需要进行绑定。在控制系统运行过程中,每个用户终端都会成为发布者,发送数据至CS组件以及其他用户终端,CS的数据也会分发至所有用户终端,因此CS与所有用户终端既是发布者,也是订阅者。这样多对多的场景下,若仍然采用直接绑定的方式,系统的复杂度会明显上升。

Xpub/Xsub模式可以避免上述问题。其核心原理是利用中间件使得Publisher与Subscriber两端解耦,并在中间件内部处理消息,从而缓解系统的压力。如图8所示,Xpub/Xsub模式中,统一由Proxy端,相当于消息转发中心,来完成绑定。Publisher与Proxy中的Xsub端连接,Subscriber与Proxy的Xpub端连接。Proxy内部,将Xsub端的消息转发到Xpub端,再由Xpub发送消息至Subscriber以实现消息的中转^[19]。

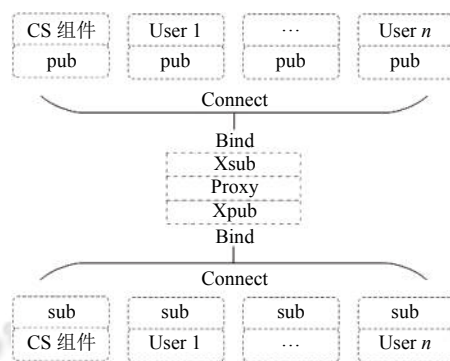


图8 Xpub/Xsub示意图

结合系统的通信吞吐量需求,选择开源的通用消息传递库ZeroMQ来实现消息转发中心。开发基于主题的发布/订阅系统,易于理解的抽象主题可以增强平台的互操作性^[20]。每个主题有对应的数据类型,界面或CS组件发布消息时,会调用各自语言版本中该数据类型的序列化接口,使消息以JSON数据流的形式在网络中传输,同理,收到已订阅主题的消息时,先调用对应的反序列化接口获得数据实体,再执行响应函数实现功能。

3.3 预设参数存储及回调

等离子体控制算法数据项繁多,每次放电前逐个设置是非常耽误时间的,因此提供回调历史炮数据至

界面中的功能,可在此基础上进行修改,这要求每次放电实验,所有的算法数据及其基本的属性都需要存储至数据库中.

图9为控制系统数据库表结构,主要有控制目录

表、算法表、波形顶点表、以及静态数据的分表,所有数据均与炮号关联,底层算法数据项如 Vertex、Parameter 等,根据“catid-phsid-subid-number”来标识所属位置.

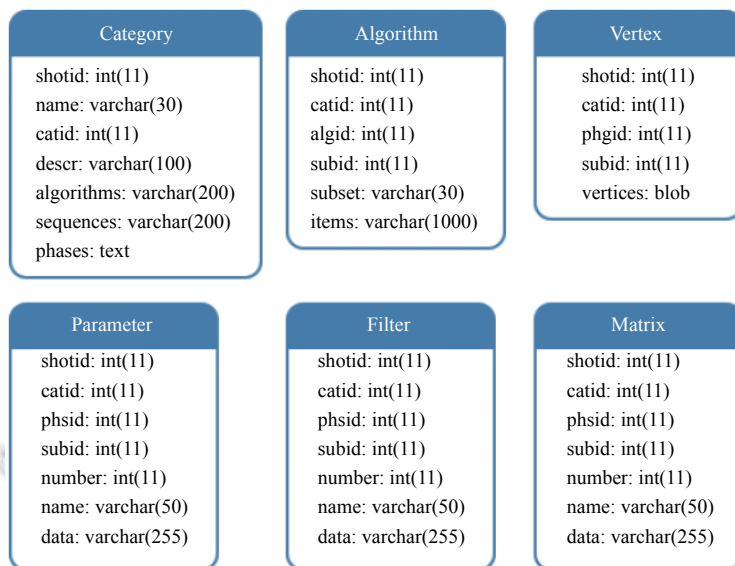


图9 数据库表结构设计

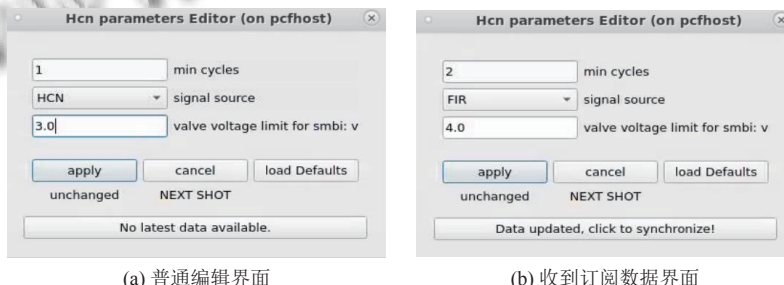
4 系统运行

操作人员在界面设置数据项的具体数值,点击“apply”按钮后,当前选中的数据便以约定好的格式(主题,数据)发布,CS组件订阅后,根据需要抽取数据进行处理,并更改数据库中对应的数值.用户界面上也会订阅这些主题,实现多人的同步修改.

以图10的Parameter界面为例,图10(a)为未收到订阅数据的编辑界面,下方按钮显示无最新数据.图10(b)中收到数据后,按钮提示数据已更新,点击便可同步.

同时根据收到数据的“category id-phase id-subset id-number”定位该项在算法树 TreeWidget 中对应的结点,并将其内置数据值替换为接收到的数据值,之后选中该项时界面便显示最新的数据.

当CS组件接收到实时调度组件的炮号信息,说明放电准备完毕,即将开始,便将当前的所有预设参数全部写入数据库,与收到的炮号绑定,同时将该数据发送至调度组件,调度组件将以约定好的方式解析、使用数据.



(a) 普通编辑界面 (b) 收到订阅数据界面

图10 多用户终端通信示例

目前CFETR暂处于立项阶段,等离子体控制系统未正式参与放电实验,因此通过采用EAST装置的控制

制算法预设参数信息进行验证和测试.已经集成实现了等离子体位形控制、密度控制以及充气控制的预设

参数编辑与管理,等磁通控制、预警控制等其他控制算法正在陆续开发中。

5 结论与展望

本文初步实现了等离子体控制系统用户界面开发,使其与具体的控制算法解耦,并且开发了统一的数据接口,基于消息订阅/发布模式实现了界面预设参数与后端组件的通信处理。

但目前版本未充分考虑多用户模式的设计实现,下一步需进行深入的研究分析。具体来说,将着重研究用户权限管理,解决多人编辑数据时可能出现的冲突问题,以及对数据请求的有效性判断。在此基础上,测试多用户数据通信的安全性以及并发性,以提高系统的性能和可靠程度。同时,将根据等离子体放电实验的实际需求不断优化界面和扩充功能,以提升用户体验。

参考文献

- 袁旗平. 基于 Linux 集群架构的等离子体控制系统 [博士学位论文]. 合肥: 中国科学院合肥物质科学研究院, 2009.
- Yuan QP, Guo HR, Yan LL, *et al.* Preliminary design of real-time plasma control system for CFETR. *Fusion Engineering and Design*, 2021, 173: 112876. [doi: [10.1016/j.fusengdes.2021.112876](https://doi.org/10.1016/j.fusengdes.2021.112876)]
- Baek S, Lee S, Park MK, *et al.* Kstar widget toolkit using QT library for the epics-based control system. *Proceedings of the 2009 International Conference on Accelerator and Large Experimental Physics Control Systems*. Kobe, 2009.
- Yin X, Zheng W, Zhang M, *et al.* A modularized operator interface framework for tokamak based on MVC design pattern. *Fusion Engineering and Design*, 2014, 89(5): 628–632. [doi: [10.1016/j.fusengdes.2014.03.028](https://doi.org/10.1016/j.fusengdes.2014.03.028)]
- Xiao BJ, Humphreys DA, Walker ML, *et al.* East plasma control system. *Fusion Engineering and Design*, 2008, 83(2–3): 181–187. [doi: [10.1016/j.fusengdes.2007.12.028](https://doi.org/10.1016/j.fusengdes.2007.12.028)]
- Harwani BM. *Qt5 Python GUI Programming Cookbook*. Packt Publishing, 2018.
- 柴文婷. EAST 放电管理系统的研究与实现 [博士学位论文]. 合肥: 中国科学技术大学, 2018.
- 张茂辉. 基于 XML 的层次化信息动态管理技术研究 [硕士学位论文]. 长沙: 国防科学技术大学, 2004.
- 信俊昌, 王国仁, 李国徽, 等. 数据模型及其发展历程. *软件学报*, 2019, 30(1): 142–163. [doi: [10.13328/j.cnki.jos.005649](https://doi.org/10.13328/j.cnki.jos.005649)]
- 陈佳铭, 王凤立, 邓君湘, 等. 基于 XML 语言的导弹防御系统 HSTPN 博弈模型存储与加载. *计算机应用与软件*, 2018, 35(12): 12–15. [doi: [10.3969/j.issn.1000-386x.2018.12.003](https://doi.org/10.3969/j.issn.1000-386x.2018.12.003)]
- 王永娜, 赵奎, 王鸿亮, 等. 针对异构协议的动态解析器模型. *计算机系统应用*, 2017, 26(1): 251–254. [doi: [10.15888/j.cnki.csa.005533](https://doi.org/10.15888/j.cnki.csa.005533)]
- 张文字, 许明健, 薛昱. 论 spring 的零配置与 XML 配置. *计算机系统应用*, 2015, 24(2): 270–275. [doi: [10.3969/j.issn.1003-3254.2015.02.052](https://doi.org/10.3969/j.issn.1003-3254.2015.02.052)]
- Nassiri H, Machkour M, Hachimi M. One query to retrieve XML and relational data. *Procedia Computer Science*, 2018, 134: 340–345. [doi: [10.1016/j.procs.2018.07.201](https://doi.org/10.1016/j.procs.2018.07.201)]
- 赵昶宇, 胡平. 嵌入式软件界面自动生成方法. *科技与创新*, 2022, (4): 141–144. [doi: [10.15913/j.cnki.kjycx.2022.04.043](https://doi.org/10.15913/j.cnki.kjycx.2022.04.043)]
- Chun WJ. *Core Python Programming*. 2nd ed. Upper Saddle River: Prentice Hall PTR, 2006.
- 徐兆桂. Python 程序缺陷的自动检测与定位技术 [博士学位论文]. 南京: 南京大学, 2017.
- Yan LL, Yuan QP, Xiao BJ, *et al.* The design of software development platform for CFETR plasma control system. *Fusion Engineering and Design*, 2020, 152: 111433. [doi: [10.1016/j.fusengdes.2019.111433](https://doi.org/10.1016/j.fusengdes.2019.111433)]
- 王开宇, 张弘瑞, 李聪, 等. 基于 IDL 通用接口的人机语音交互协同控制系统设计研究. *无人系统技术*, 2021, 4(4): 77–84. [doi: [10.19942/j.issn.2096-5915.2021.4.039](https://doi.org/10.19942/j.issn.2096-5915.2021.4.039)]
- 吴昊. 基于 NNG 的高速数据分发系统设计与实现 [硕士学位论文]. 兰州: 兰州大学, 2019.
- 杨勇, 薛化建, 任鸽. 基于发布/订阅模型的异构网络数据交换方法. *计算机系统应用*, 2020, 29(7): 103–109. [doi: [10.15888/j.cnki.csa.007494](https://doi.org/10.15888/j.cnki.csa.007494)]

(校对责编: 孙君艳)