

故障信息标准化描述及网络传输方法研究

陈希祥, 申宇皓

(湖南信息学院 电子信息学院, 长沙 410007)

摘要: 为满足网络故障诊断中跨平台故障信息标准描述和网络传输的需要, 基于 LabVIEW 的跨平台和 XML 的标准语言描述能力, 开展相应的详细设计与实现方法研究, 给出典型故障的 XML 描述方法; 通过 TCP 协议, 借助局域网中的客户机/服务器模式来实现故障信息的网络传输; 在服务器端, 采用 VI 脚本技术动态创建和运行软件模块, 以满足不同故障信息传输和处理的需要; 在客户端, 供应商和客户模式被用于增强安全性以避免数据丢失; 构建的仿真系统验证了设计的可行性, 研究成果对今后进一步实现实用的网络化故障诊断系统具有一定的参考价值。

关键词: 故障信息描述; LabVIEW; XML; VI 脚本

Research on Implementation Method for Fault Information Standardized Description and Network Transmission

Chen Xixiang, Shen Yuhao

(School of Electronic Information, Hunan University of Information Technology, Changsha 410007, China)

Abstract: The detailed planning and implementation method are introduced based on LabVIEW for the sake of satisfying the need of cross platform, fault information standard description, and network transmission in networked fault diagnosis. Relevant research was based on the capacities of LabVIEW cross platform and of XML standard description. The XML description method of typical fault information was given. Network transmission of fault information was realized by means of the client/server mode in a LAN through TCP protocol. The technique of VI Scripting was adopted to create and run software module dynamically in the server to meet the need of transmitting and processing different fault information in future. In the client, producer and consumer mode was used to enhance the security to avoid the loss of data. A constructed simulation system verified the feasibility of planning. The research production is useful for implementing practical networked fault diagnosis system further in future.

Keywords: fault diagnosis information description; LabVIEW; XML; VI scripting

0 引言

随着科学技术发展, 现代工业系统的模块化、网络化、数字化、智能化水平有了很大提升, 设备结构也越来越复杂, 系统内部存在着复杂关系。系统功能单元故障原因复杂, 故障表现出多样性、渐变性、耦合性等特点^[1-3]。传统的现场检测与诊断模式已不能满足快速性、实时性等实际需要, 开发网络化、信息化故障诊断系统是工业系统故障诊断的发展趋势^[4]。网络化故障诊断系统通过传输和处理现场故障信息, 将现场人员与远程专家相结合, 利用不同的领域知识处理系统, 提高故障诊断的准确性和效率。

网络化故障诊断系统的实现包括跨平台、故障信息标准描述、网络传输等关键基础技术。跨平台技术可提高诊断系统在不同操作系统中的部署能力及针对不同开发平台的适应能力, 提高检测诊断系统的通用性; 故障信息标准描述增强了不同开发平台之间的信息共享能力; 网络传输是实现故障检测信息与诊断决策信息远程传输及交互的关键。

在不同研究领域, 对诊断方法都进行了大量的研究和讨论^[5-7], 然而所开发的系统在跨领域、跨平台和故障信息标准描述方面显得不足, 通用性较差, 难以实现远程信息分析与故障诊断。通过对比分析, LabVIEW (laboratory virtual instrument engineering workbench) 可支持跨平台系统的快速开发^[8-10], 而 XML (Extensive Markup Language) 也支持跨平台信息描述, 独立于硬件和软件。将 LabVIEW 和 XML 应用于网络化故障诊断系统的开发, 可以使系统具有很强的灵活性和通用性^[8-12]。因此, 本文选择 XML 实现故障信息标准描述, 以 LabVIEW 作为开发平台实现系统构建及信息网络传输, 从而实现跨平台网络化远程故障诊断系统。文献 [11] 中给出了在 LabVIEW 中创建和访问 XML 文件的基本方法。但是, 它对不同类型的信息缺乏灵活性。文献 [12] 中讨论了在 LabVIEW 中实现远程数据传输的基本方法。然而, 它不能避免数据的丢失。

本文首先给出了系统总体规划, 详细介绍了设计过程, 然后给出了服务器端和客户端程序的详细设计, 用 XML 描述典型故障模式影响分析信息 (FMEA, failure mode effect analysis)。利用 VI 脚本技术实现程序的动态创建和运行, 采用 TCP 协议实现数据传输, 采用供应商和客户模式增强了安全性, 避免了数据丢失。最后构建了一个演示仿真系

收稿日期:2020-03-09; 修回日期:2020-04-15。

作者简介:陈希祥(1981-),男,山西祁县人,博士,工程师,主要从事机电一体化、工业自动化、状态监控与故障诊断方向的研究。

统, 讨论仿真试验结果, 对本文进行总结。

1 总体方案设计

本节从跨平台、故障信息标准描述、网络传输等方面探讨如何选择和优化相关技术, 实现总体设计。

1.1 确定开发平台

跨平台是软件开发中一个重要的概念, 是指程序设计语言、应用软件或硬件设备可在多个操作系统中工作。由此可见, 跨平台既不依赖于操作系统, 也不依赖硬件环境, 可提高软件的重用性。

JAVA 是 SUN 公司推出的 Java 面向对象程序设计语言和 Java 平台的总称, 它利用虚拟机技术实现跨平台, 但开发效率不高。LabVIEW 是由美国国家仪器 (NI) 公司推出的重要软件产品, 它是一种图形化编程语言, 完全支持跨平台, 其代码无需更改即可运行于 3 种常见的操作系统: Windows、Mac OS 和 Linux。同时, LabVIEW 也是一种程序开发环境, 可实现系统快速的开发。根据大量统计数据发现, 开发同一个大型应用程序, LabVIEW 程序员只需花费 C 程序员 1/5 的时间。因此, 本文选择 LabVIEW 作为开发平台实现网络化故障诊断演示仿真系统。

1.2 信息描述语言及其实现方法

XML 是一种基于文本、使电子文件或信息具有结构性的标记语言, 它将可理解的标记和结构化格式应用于保存数据。XML 是支持跨平台、依赖于内容的技术, 提供统一的方法来描述和交换独立于应用程序或供应商的结构化数据, 已经成为数据交换标准, 非常适合网络传输, 是当今处理分布式结构化文档信息的有力工具。

相比于 HTML 有一个固定的标记集, XML 只提供一个标准, 用户可按照这个标准来定义自己专用的标记, 所以 XML 的标记是可以无限扩展的, 可用于描述各种应用领域的的数据信息, 人们可以理解并且计算机可以处理它。因此, 选择 XML 来描述故障信息。

在 LabVIEW 中实现信息的 XML 描述有两种方法: 一种是 LabVIEW 模式, 另一种是 XML 解析器。LabVIEW 模式根据预定义的 XML 模式转换数据。简单的几个函数可以完成基本的 XML 操作。XML 解析器是 Xerces 2.7, 它依赖于 DOM (文档对象模型), 提供了许多灵活处理 XML 数据的功能, 但它不支持除英语以外的其他语言。因此, 选择 xerces 2.7 模式来实现故障信息的 XML 描述, 同时, 采用了一些技术措施来提高 Xerces 2.7 模式的灵活性。Xerces 2.7 模式中 FlattedToXML 函数给出字符串的 XML 转换结果如下:

```
<String>
<Name>Function</Name>
<Val>Propellant Storage</Val>
</String>
```

其中 <String> 标记表示输入数据类型为字符串“String”。<Name> 标记表示输入的名称 (如功能 function)。<Val> 标签表示功能内容 (如燃料储存)。

1.3 常见故障信息及其动态描述方法

故障信息的类型和形式是多种多样的。FMEA 是诊断中故障信息的重要来源, 对典型故障信息 FMEA 进行描述和传输, 实现总体规划设计的演示验证。通常 FMEA 信息用 FMEA 表来表示, 明确了系统或设备故障的详细信息, 包括名称、功能、故障模式、故障原因、故障影响 (局部、进一步、最终影响)、故障检测方法等。

在设计程序时, 要能够对标准的 FMEA 信息进行处理, 并使其具有可扩展性。通常 FMEA 表信息的字段名和内容不是固定的, 本文采用动态的方式实现了 LabVIEW 中常用的表信息的 XML 描述, 将 VI 脚本技术应用于创建 VI、控件、函数、运行程序以及结果输出。VI 脚本依赖于本地 LabVIEW VI 服务器所提供的的服务, 客户端则是本地 LabVIEW 程序。

1.4 网络通信协议选择

TCP 和 UDP 是两种重要的网络通信协议。TCP 定义了两台计算机之间可靠的数据传输格式和方法, 其重要特性是提供面向连接的可靠字节流服务。UDP 是一种简单的面向数据包的协议虽然 UDP 的传输速度较高, 但其提供的传输服务并非面向连接且是不可靠的。因此, 选择 TCP 作为演示仿真系统的通信协议实现故障信息的传输需要高可靠性, 由服务器描述并发送信息, 客户机接收和处理信息。

如果客户机以顺序循环的方式接收和处理信息, 数据将不能被及时接收, 且由于处理数据花费了大量时间, 会发生数据丢失。该模式采用队列的数据存储方式, 按先进先出 (FIFO) 方式工作, 新元素总是加载队列的末端, 而最先取出的数据则是最先进入队列、位于队列首段的数据。因此以该方式进行数据的存取可避免数据遗失。供应商和客户模式利用多个环路并行实现供应商和客户的不同功能, 可提高数据传输的安全性。

1.5 总体规划设计

通过以上分析, 对演示仿真系统进行规划设计。该仿真系统由服务器端和客户端组成, 服务器实现故障信息 FMEA 的 XML 描述, 并将描述结果传送到网络; 而客户端与服务器连接, 用于接收服务器传送的网络数据, 提取并显示故障信息。

服务器端和客户端的工作流程分别如图 1、图 2 所示。在服务器中, 首先创建一个网络连接, 利用 XML 对故障信息进行描述, 并发送到网络中, 直到信息全部发送完毕。然后服务器等待, 直到客户端接收到最后一组数据信息为止, 并关闭网络连接。在客户端, 首先打开网络连接, 接收网络数据、提取并显示故障信息。服务器和客户端按一定时序同时工作, 确保数据信息的发送与接收同步进行, 交互的工作流情况如图 3 所示。

仿真系统技术框架如图 4 所示。如前所述, 整个系统采用 LabVIEW 开发, 支持跨平台。服务器通过 XML - LABVIEW 模式和 VI 脚本技术, 利用 XML 实现典型故障信息 FMEA 的描述, 客户端根据供应商和客户模式技术接

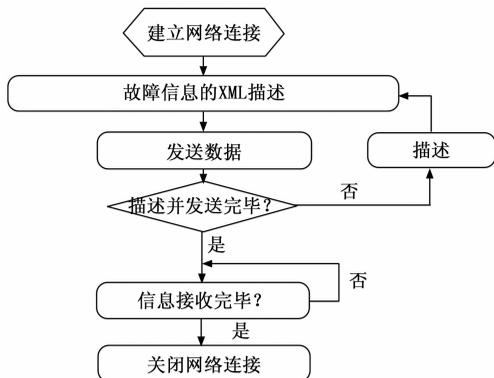


图 1 服务器 workflow

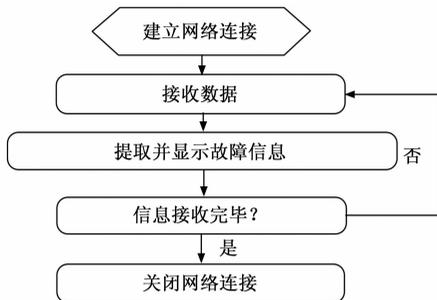


图 2 客户端 workflow

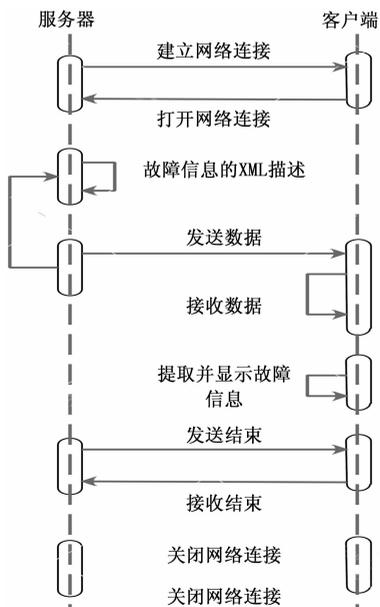


图 3 服务器与客户端的交互 workflow

收网络数据, 提取并显示传输的故障信息, 两者之间通过 TCP 协议进行通信。

2 具体实现方法

本节对总体方案中的关键技术及其实现方法进行探讨。

2.1 服务器

服务器控制面板如图 5 所示。“FMEA”表显示将要发送的故障信息, “Sending XML”文本框显示 FMEA 表中数

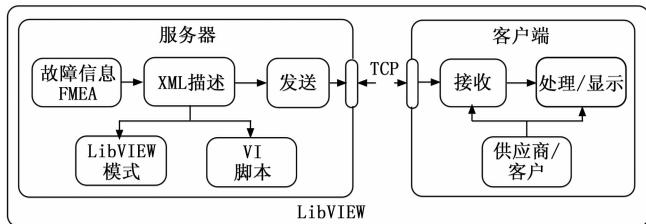


图 4 仿真系统技术框架

据信息的 XML 描述结果, “Port” 端口控制用于输入 TCP 通信的端口号, “Sending” 指示器用于显示发送状态, “Start” 启动按钮用于运行程序。服务器利用 XML 来描述 FMEA 的数据信息, 并逐条将描述结果发送到网络。

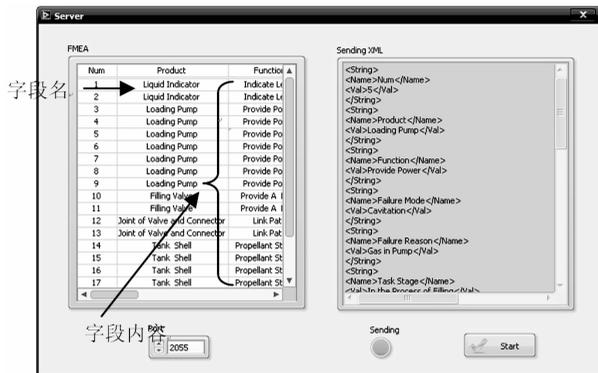


图 5 服务器面板

服务器框图如图 6 所示。框图的外部是响应启动按钮 “Start” 动作的循环事件结构。图的内部由 5 个模块组成: 1) 创建 TCP 连接; 2) XML 描述和发送数据; 3) 发送 “发送结束” 字符串; 4) 等待 “接收完成” 字符串; 5) 关闭 TCP 连接。

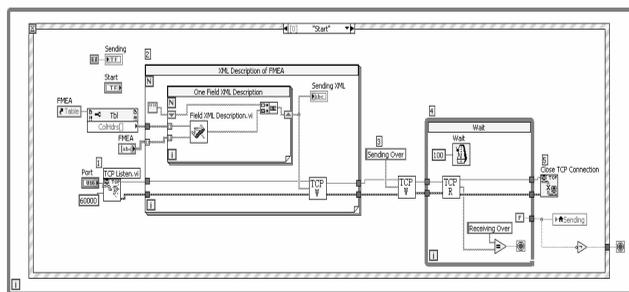


图 6 服务器框图

在第一个模块中, 程序创建一个侦听器, 等待来自客户端端口上的 TCP 连接, 等待时间为 1 分钟。在第二个模块中, FMEA 表是数据源, 由 XML 逐条记录、逐个字段分别描述, 在一条记录被描述并发送到网络之后, 开始描述下一条记录。

为满足不同故障信息描述的需要, 在 LabVIEW 中采用 VI 脚本技术描述 FMEA 记录中的某一字段。此时, 需要创建一个新的 VI, 如图 7 所示, 以便每个字段的描述结果与第 1.2 节 LabVIEW 模式中的示例格式保持一致。面板由两

个文本框组成，一个用于输入字段名和字段内容，另一个显示该字段的描述结果。相应的框图由输入文本框的局部变量和 Flatted to XML 函数组成。



(a) 面板 (b) 方块图

图 7 通过 VI 脚本创建 VI

为创建上述 VI，采用 VI 脚本技术设计一个新的 VI：Field-XML 描述。Field-XML 描述框图由“创建 VI”、“创建输入控制及局部变量”、“创建函数和连线”、“创建输出控制和连线”、“运行 VI 及获取结果”等 5 个模块组成，如图 8 所示。“创建 VI”即创建一个新的面板和一个新的框图。“创建输入控制及局部变量”是为了创建一个文本框，包含面板中的字段名和字段内容，以及框图中的局部变量。“创建函数和连线”用于创建 Flatted to XML 函数，并将该函数与输入文本框的局部变量连接。“创建输出控制和连线”用于在面板中创建输出文本框，并将输出文本框与 Flatted to XML 函数连接。“运行 VI 及获取结果”为了使创建的 VI 运行并得到最终结果。

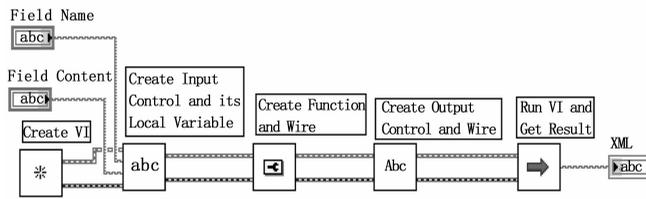


图 8 VI 字段 XML 描述框图 vi

通过 VI 脚本设计 VI 是很复杂的。在 2 GHz Intel i5 CPU、2 G 内存、Windows XP、LabVIEW 2012 环境下运行 VI 将花费更多的时间（约 80 ms）。而 VI 脚本实现了 VI 设计的自动化，提高了软件的灵活性，可满足不同需求。Field-XML 描述 VI 使用了许多关于 VI 脚本的属性和方法。鉴于篇幅有限，本文对这些性质和方法未进行详细讨论。

服务器框图中的模块 2~4 包括发送网络数据和读取网络数据。为方便起见，创建两个 VI 分别实现上述两个功能，一个是 TCP W (Write)，另一个是 TCP R (Read)。在这两个 VI 中，读写的字符串长度需要输入。

在 FMEA 信息的 XML 描述全部发送后，在第 3 个模块中，通过发送“发送结束”字符串帮助客户端识别发送端。在第四个模块中，客户端收到所有数据后，服务器等待从客户机发来的“接收结束”字符串。最后，关闭 TCP 连接。

2.2 客户端

客户端从网络逐条接收 XML 记录，提取并显示该记录

所包含的故障信息，客户端面板如图 9 所示。“Receiving XML”文本框用于显示接收到的 XML 描述记录；“XML File”文本框用于显示保存记录的 XML 文件；“FMEA”表用于显示从 XML 描述中提取的故障信息；“Port”端口控制用于输入 TCP 通信的端口号；“Address”地址控制用于确定网络连接地址，其内容是连接客户端的本地计算机名称。由于客户机使用包含队列的供应商和客户模式，因此“No. of Elements in Queue”一栏显示队列中的元素数；接收指示器“Receiving”显示接收状态；启动按钮“Start”用于运行程序。

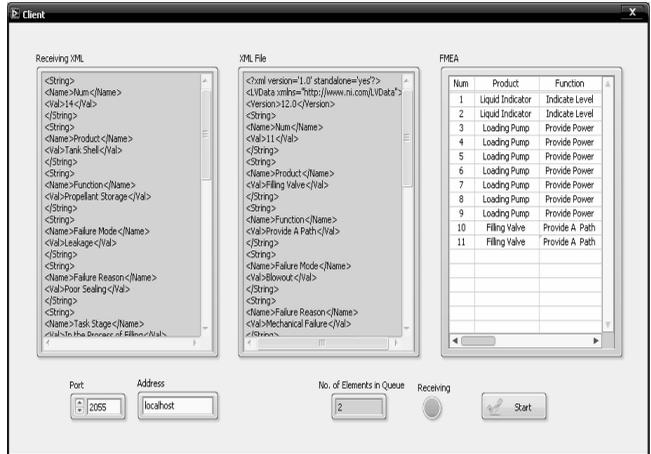


图 9 客户端面板

客户端如图 10~11 所示，包括供应商和客户两个独立的循环。供应商循环负责接收网络数据并将数据放入队列，客户循环则负责从队列中提取数据及故障信息。当接收和处理数据的工作是按顺序周期性进行时，由于数据处理时间很长，因此，在供应商和客户模式下数据不完全接收的情况是一般不会发生的。

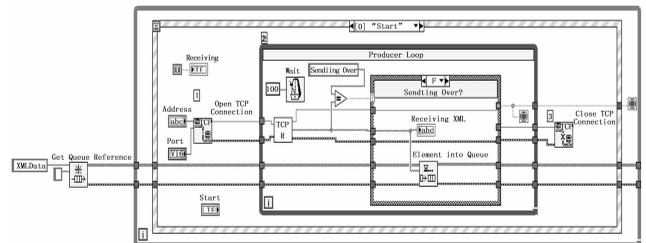


图 10 供应商循环

供应商循环的外部是响应开始按钮“Start”动作的循环事件结构。该结构运行之前，需要为后续队列操作创建队列引用，队列名称是 XML 数据，队列中的元素类型为字符串。循环事件结构框图包括 3 个模块：1) 打开 TCP 连接；2) 接收 XML；3) 关闭 TCP 连接。要打开 TCP 连接，需要端口号和地址，随后，开始接收来自网络的数据。如果接收的数据没有发送完，则数据将显示在接收 XML 文本框“Receiving XML”中，并插入到队列，以便客户循环从队列中提取故障信息。当数据接收完毕时，表明服务器已

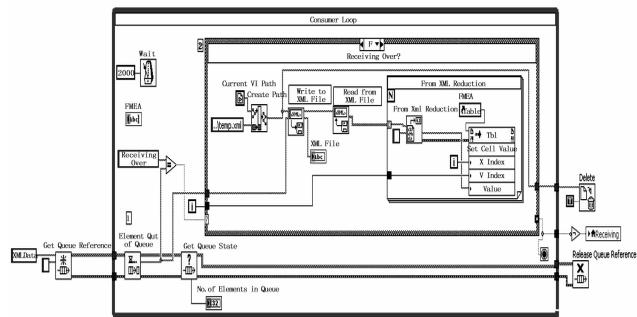


图 11 客户循环

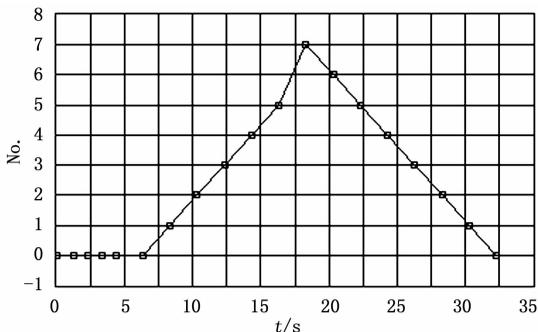


图 12 信息传输队列中的元素数

将所有数据发送完毕, 客户端将项服务器发送“Receiving Over”信号。同时, 该字符串将被插入到队列, 使用者退出程序, 关闭 TCP 连接。若服务器接收到“Receiving Over”信号时, 服务器退出程序, 而不必等待客户端处理数据。

在客户循环中, 队列中的元素被逐一提取。如果提取的数据未接收完毕, 则持续提取故障信息。首先, XML 数据被保存为一个“*.xml”文件, 以供客户读取。所读取的数据为数组, 元素格式与第 2.2 节中示例相同, 这样通过数组操作即可提取故障信息。然而, 从“Receiving XML”文本框所示的原始数据中提取信息并不容易, 因此可创建一个临时文件“temp.xml”来保存数据。由于需要保存和读取文件, 客户的一个循环周期很长, 如果采用更复杂的方法来处理数据, 循环周期将更长。当数据提取结束时, 程序退出客户循环, 释放队列引用, 并删除临时文件。

2.3 仿真结果

假设在服务器中需要传输 20 条故障信息记录。服务器和客户端程序在本地计算机上运行, 端口号为 2055。首先运行服务器, 客户端随后运行, 工作正常, 其 workflow 如图 5 和图 9 所示。当程序运行一次时, 客户端队列中的元素数量如图 12 所示。开始时数据量是 0, 接收到的数据可被及时处理。随后, 数据量逐渐增加。在 18 s 后达到最大值 7, 此时服务器上的所有数据都已完全发送, 然后对队列的元素逐个进行处理, 数据量逐渐减少。在 2 GHz Intel i5 CPU、2 G 内存、Windows XP、LabVIEW 2012 环境下, 整个程序大约运行花费约 32.5 s。

由仿真过程可以看出, 采用 XML 进行故障信息标准化描述, 为实现故障信息跨平台远距离传输与共享奠定了基础; 采用客户端—服务器的网络架构可有效实现信息传输, 有助于设备故障的远程监控与诊断, 增强故障诊断的实时性与灵活性。仿真结果与预期一致, 进一步验证了总体方案与技术的可行性。

3 结束语

为满足网络故障诊断中跨平台、故障信息标准化描述、网络传输的需要, 基于 LabVIEW 给出了设备故障信息标准

化描述及网络传输总体设计方案和实现方式。通过采用多种技术, 验证了总体技术的可行性。本文研究成果有助于进一步实现机电一体化设备远程跨平台网络化故障诊断系统和 LabVIEW 的相关开发工作, 为开展复杂故障信息的描述与传输研究工作提供了思路。

参考文献:

- [1] 郑旭, 李亚光, 邱辉. 工业机器人故障分析及可靠性测试的必要性 [J]. 中国仪器仪表, 2019 (10): 29-33.
- [2] 陈乐瑞, 曹建福, 王晓琪. 工业机器人驱动系统非线性频谱故障诊断方法 [J]. 西安交通大学学报, 2019, 53 (4): 93-99.
- [3] Wen L, Li X, Gao L, et al. A new convolutional neural network-based data-driven fault diagnosis method [J]. IEEE Transactions on Industrial Electronics, 2018, 65 (7): 5990-5998.
- [4] 王海峰. 基于工业大数据的智能化能耗管理与故障预诊系统 [J]. 电气自动化, 2018, 40 (4): 109-110.
- [5] 黄政庭, 王仲生. 基于 DFT 的飞机电源网络化故障诊断方法 [J]. 计算机工程, 2011, 37 (24): 233-235, 238.
- [6] 杨帆. 基于 C/S 架构的旋转机械故障诊断系统设计 [D]. 重庆: 重庆大学, 2016.
- [7] 徐启圣, 白琨, 徐厚昌, 等. 风电齿轮箱状态监测和故障诊断的研究现状及发展趋势 [J]. 润滑与密封, 2019, 44 (8): 138-147.
- [8] Wang J Z, Chen M Z, Jin S, et al. Data communication mechanism for greenhouse environment monitoring and control: An agent-based IoT system [J]. Information Processing in Agriculture, 2019.
- [9] Shokri M, Mirabi M. An efficient stream structure for broadcasting the encrypted XML data in mobile wireless broadcast channels [J]. The Journal of Supercomputing, 2019, 75 (11): 7147-7173.
- [10] Nazariy A, Ivan K. LabVIEW-based automated setup for interferometric refractive index probing [J]. SLAS Technology, 2019.
- [11] 罗锡梁, 杨世锡, 蔡清华, 等. LabVIEW 中创建和访问 XML 文件的实现 [J]. 现代机械, 2006 (2): 62-63, 78.
- [12] 丁硕. 基于 LabVIEW 的远程数据通信技术的实现 [J]. 电子科技, 2008 (5): 42-44, 48.