

开放式控制器开发中的软件质量保证体系

赵英普¹, 陈 虎¹, 穆 欣¹, 韩至骏², 张 昆²

(1. 北京首科凯奇电气技术有限公司, 北京; 2. 清华大学 机械工程学院, 北京)

摘 要: 阐述了软件质量保障体系对开放式控制器开发的重要性。在分析开放式控制器软件基本特点和目前控制器软件开发普遍存在的问题的基础上, 提出了控制器软件质量保障的一些基本措施。

关键词: 数控技术; 开放; 软件工程; 质量管理

中图分类号: TP31

文献标识码: B

文章编号: 1009-0134(2005)05-0021-04

The software quality control system of development of open architectural controller

ZHAO Ying-pu¹, CHEN Hu¹, MU Xin¹, HAN Zhi-jun², ZHANG Kun²

(1. Beijing Shouke CATCH Electric technology Co., Ltd, Beijing; 2. Tsinghua University, Beijing)

Abstract: The paper discusses importance of software quality control system for open architecture controller. After analyzing the character of controller software and the problem in development, some steps are given to ensure the quality of controller software.

Key words: CNC; open architectural; software engineering; quality control

0 引言

在机床控制器产品及其相关技术领域, 开放化已成为一种潮流。美国有NGC项目, 欧洲的OSACA计划, 日本也有OSEC计划。主流的控制器的厂商如Siemens也在自己的产品中提供了具有开放特色的功能以供客户在VC和VB上开发自己的人机界面。我国的标准化组织也一直致力于成立自己的开放式控制器标准, 组织国内诸多控制器厂商、高校和科研院所共同制定了GB/T 18759.1《机械电气设备 开放式数控系统 第1部分: 总则》标准。凯奇公司也是积极参与该标准制定的单位, 同时以标准为依据, 指导新一代控制器产品的开发。是对我们在开放化控制器软件开发过程中对于软件质量保证体系的总结。

1 软件可靠性是开放式控制器可靠性的关键

目前, 数控系统的硬件体系结构主要有以下几种:

- 基于PC硬件体系结构的控制器。
- 前后台系统, 一般前台是通用的PC架构, 后台使用嵌入式系统完成控制器的实时核心。
- 由以DSP为核心的专用硬件体系结构

随着PC制造业的迅猛发展, 基于PC硬件体系结构的控制器具有扩展性、兼容性强, 制造成本低, 产品可靠性高, 硬件体系结构中运算系统升级方便, 为控制器复杂算法提供了良好的运算资源。PC架构中包括的操作系统资源也为控制器软件开发提供了良好的开发环境。PC硬件体系结构的开放性, 也为控制器产品的开放化奠定了良好的体系结构基础。因此, 可以说上述3种硬件体系结构中基于PC硬件体系结构的控制器因其对开放化的支持, 可能最富有生命力。

另一方面, 由于PC技术的同质化, 导致基于PC的控制器在硬件上也趋于同质化, 产品功能和性能的竞争的重点已经从硬件转到软件层面上。这本身也是控制器追求开放化, 追求基于开放化控制平台上的控制器个性化的原动力之一。控制器软件在控制器组成中的作用是举足轻重的。控制器整体内在品质和可靠性很大程度上与控制器软件系统的可靠性密切相关, 特别是当控制器硬件可靠性问题日臻改善的时候, 软件可靠性的问题就更加突出。因此, 软件可靠性是开放式控制器可靠性的关键。

2 开放式控制器软件系统的特点

数控系统软件由逻辑上分离的组件所组成, 如

收稿日期: 2005-01-07

作者简介: 赵英普, 北京首科凯奇电气技术有限公司。

插补计算软件模块、PLC 控制软件模块等。开放式的终极目标是来自于多个供应商的组件可以协同工作在不同的平台上。目前的开放式计划只能说是一种准开放,也就是开放系统的一部分接口向高级用户或最终用户开放,用户在此基础上进行二次开发。尽管如此,仍然要求软件系统满足开放性的一些基本要求,包括自身的健壮(对非正常运行状态的自行处理能力)和对外接口的一致性。这样才可以保证软件系统自身不会成为系统可靠性方面最薄弱的环节。因此在系统平台的开发中必须建立严格的软件质量保证体系,同时应尽量在第3方的开发中贯彻认证和审核体系,以保证整个系统的可靠性。

控制器软件系统区别于一般的软件,在于控制器软件中包含有强实时性要求的算法模型。这部分代码除了普通代码正确性要求外,具有严格运算资源开销限制(包括运算时间和存储空间开销)。在评价控制器软件时,这部分模块的评价准则比其他代码更加严格。

3 控制器软件开发中经常遇到的问题

我国关于数控技术的研究历史较长,可以追溯到上个世纪50年代。但中国的控制器产业发展并不尽如人意。控制器产业发展不顺利的原因是多方面的。从大环境看,控制器软硬件技术从80年代中期到现在是发展进化最快的历史时期,而我国的产业界正处于体制转型,与市场接轨时期。在这个时期,包括人才、市场、相关产业的诸多问题影响了企业的研发工作进度。另一方面,不仅是控制器产业,很多产业都是刚刚接受软件工程化思想。如何像制造其它产品那样,使软件的生产走向规模化,是此前控制器产业界未曾遇到过的问题。上述问题导致我们在控制器软件“制造体系”方面的工作显得比较苍白。

具体地讲,目前控制器厂商关于控制器软件开发中普遍存在的问题:

- 系统的软件开发由硬件人员承担。这种情况对于业界的“经济型”控制器产品是普遍存在的。由于“经济型”控制器产品的硬件体系是以特定处理器为核心的专用硬件体系结构,该结构上也大多没有操作系统,软件工作也被限定在特定的运算资源下。这种情况下,硬件、软件、模型设计大多均由一个研发小组完成,仍是典型的“自编、自导、自演”小作坊的工作方式。这种研发模式距离开放式控制器的多供应商提供软件部件,互易操作的目标

相差很远,软件的质量也是不确定的。

- 软件人员不太习惯于软件工程化、规范化、结构化和模块化的软件生产方法,往往跳过了软件设计阶段,先有编码,为了软件检查才补设计。这一点几乎是所有企业软件生产过程中的通病,是实施软件工程前普遍存在的状态。这种情况下软件质量(稳定性和效率)并没有在设计阶段被充分考虑,软件质量最终也无法得到保证。

- 缺少配套的软件测试工具。由于无法对模块级算法进行测试(大多数情况是由于模块划分不够清晰造成),试图最终利用目标实时控制系统硬件进行软件的调试、测试、验证、确认和试验工作。这样的软件测试应当属于黑盒测试,很难保证测试用例设计的完整,是有局限性的。

- 缺乏对实时控制系统软件系统可靠性的深入研究。这些研究包括详尽深入地了解操作系统平台的实时性(具体化到操作系统任务调度的开销、内存分配的开销、外设操作的开销和中断响应过程的开销等),也包括对控制算法本身每个计算环节和数据交换环节开销的把握。无法得知系统的行为细节,自然无法控制系统的计算资源,系统的运算负荷合理分配自然无从谈起,实时控制软件的质量也就无法得到保证。

4 开放式控制器软件质量保障

开放式控制器软件是一项严谨而庞大的软件工程,它相对于其它软件工程除了具有特定的应用领域和机械制造领域独特的使用需求外,还格外强调系统软件的实时响应能力。我们在实际研究工作中,关于开放式控制器软件质量保障有以下几点体会:

4.1 开放式控制器软件质量首先是设计出来的,而不是单纯的保障

纵观国际上著名的开放式控制器研究计划,都能看到清晰的体系结构,严谨的接口描述为软件工程化实施奠定良好的基础。美国NGC计划中最瞩目的就是将控制器软件的需求抽象为若干组应用程序接口(API),严格约定了API的功能和接口。欧洲开放式控制器研究计划OSACA更是详尽而完整地约定了软件体系结构的模块划分;将软件建模的方法统一到采用面向对象技术;在中间件技术尚不成熟的情况下,详细约定了模块间数据交换机制和配置机制。这些研究计划除了在技术上给我们启发外,其研究方法和组织实施的步骤都给我们今天的软件

开发以启示。软件体系的可靠性是需要在设计阶段关注的。

控制器软件设计阶段能够在以下几个方面对最终控制软件的质量起到积极作用：

- 合理的模块划分和清晰的接口定义能够将整个软件的质量控制目标分解为各模块质量控制指标，使质量控制更加具体化。

- 设计阶段要对硬件平台和操作系统的相应实时性进行评估，针对系统软硬件平台设计模块的调度策略。这是保障系统实时性的基础。

- 对于模块级的设计，需要考虑模块可能遇到的异常状态，并给出处理异常的专有接口。这样就将许多可能的故障在设计阶段就处理了。

- 在模块级设计工作的同时，考虑模块级的测试设计。从模块外部的需求出发，同步设计测试用例，一方面减轻了最终设计测试用例的负担，另一方面也可以换个角度对设计工作可能的错误及时校正。

4.2 重视软件开发工具

软件主要的生产工具就是软件开发的平台，一般包括操作系统平台和相应的开发语言，以及相应的辅助工具。对于控制器软件非实时部分，除了可靠性以外，质量的重点在于更好的模块性及可扩展性，这也是开放化所重视的软件特质。因此在开发中须选择具有较强的移植能力和具备一定的模型描绘能力的程序设计语言，例如当前使用较多的是C/C++语言。

鼓励使用标准库。使用已经过业界验证的开发库，可以减少开发团队编写程序的代码量。根据软件工程的原理，代码的错误是和代码量成正比的。一般业界经过验证的库都是由水平很高的程序员书写，并经过大量的实际应用测试，大量实践证明这些代码可以达到令人满意的质量。在选用标准库时，要注意选用移植性较好的库。有时为了提升性能而选用大量具有特定平台特征的库，将导致系统丧失可移植性，这与开放化趋势也是背道而驰的。

对于系统中的实时部分，可靠性和代码的效率是首先需要考虑的。在语言选择上优先选用的C和汇编。在开发过程要经常使用工具对程序作性能测试以及Profile和代码覆盖检查，以找出程序中耗时最多的地方，并加以改进。在主要的开发平台中都有类似的工具来实现以上功能。此外严格的性能测试可能需要精确的计时方法，比如记录CPU的时钟

计数以及使用专用的硬件设备来计时。

4.3 强化软件测试环节

软件测试对于软件质量的保证是非常重要的。软件测试是贯穿于开发的整个过程中的。应解决这样一个认识问题——用于实时控制系统一类的复杂软件，认为没有错误的想法是不切合实际的。因此，测试的主要目的是：

- 对软件的质量或可接受性做出判断
- 发现各种问题，并使之获得解决

图1展示了测试过程与软件开发的各个环节间的互动关系。

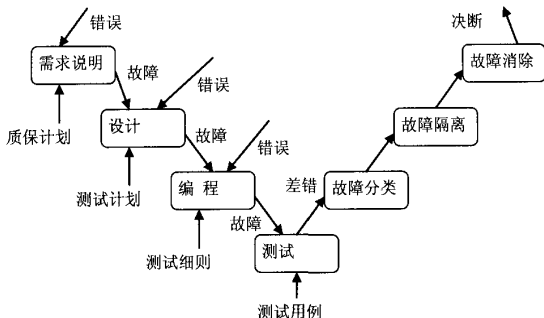


图1 测试参与系统软件设计的各个环节示意图

对于测试来说，从需求分析开始就应该着手建立对应的测试计划，在设计和编码阶段，逐渐完善并执行整个测试计划。如果测试没有发现的错误在工程完成后再修改的话，要付出当初数倍乃至数十倍的时间。在整个工程的过程中，可能付出30%-40%的时间用于测试。

测试并不仅仅指整个工程完成后的功能测试，还包括代码阶段的单元测试，以及之后的回归测试。此外，建议建立一个测试管理系统，搭配某种自动测试工具。首先是单元测试，主要是针对编码过程中在某一个逻辑单元中添加某些测试代码，在测试框架的驱动下所完成的测试，比如C和C++有相应的CPPUnit测试工具，在发现问题并修正后要进行回归测试，以确定在修正错误的同时没有增加更多的错误。在单元测试完成后，要进行功能测试，在整个过程中可以使用专门的自动化测试工具，对于GUI来说还有专用的工具可以模拟界面操作。此外要建立相应的测试数据库，来记录测试用例及测试结果，如果发现问题要更正记录错误的解决办法以及回归测试的结果。对于实时系统因为其时间依赖性以及模块间严格的同步关系还要进行综合性测试，包括任务测试、行为测试、任务件测试和系统测试。

对于某些对可靠性要求较高的模块要做相应的强化测试,以验证极端情况下的性能和稳定性。

软件产品最终的功能和性能综合测试环节也是必不可少的。通常,最终测试又可分为公司内部测试和公司外部测试。在最终综合测试环节中要设计所有的可能操作使用的实例对软件系统全面测试。在外部测试中允许用户进行非常规的操作,从而发现软件系统潜在的,超出设计者思维方式的问题。

4.4 不可或缺的评审环节

对于控制器产品,最终的评审是用户和市场。作为控制器厂商当然希望市场的评审意见都是正面的。为了在市场上有良好评价,控制器产品出厂前应当有内部的评审环节,软件生产也是一样。此处特意强调评审环节,是因为评审环节不仅对于软件质量,而且对于软件产品的改进、系列化和长期发展有重要的意义。

软件评审是软件生产过程中过滤软件错误的一个“过滤器”。软件评审涉及评审的组织机构、管理、准则、类别、内容、文件和要求等。一般要求在软件研制阶段的里程碑点进行软件评审。评审的主要类别有:软件定义评审、软件需求评审、概要设计评审、详细设计评审、软件实现评审和软件验收评审等。

以下原则是在软件评审中需要被遵守的:

- 某阶段未通过阶段评审不得进入下一个软件研制阶段;
 - 评审时对事不对人,评审的是产品,而不是评审生产者;
 - 评审就要挑剔,找问题、缺陷和隐患;
 - 评审组的人员面越广越好;
 - 评审组不作无休止的争论和辩驳,将争论点记录下来,供以后甄别;
 - 评审只是提出问题,没有解决问题的任务;
 - 使用“评审检查单”以提高评审的效果;
- 经过评审环节,我们可以达到以下目的:
- 技术把关。通过评审可以及时纠正软件开发

人员设计思想上的偏差,可以避免软件人员的想当然。评审保证了软件开发始终沿着既定技术路线进行。

● 概念沟通。评审过程吸收用户和总体人员参加,通过多个角度审查软件人员理解的正确性。许多需求分析中被忽略的技术边界条件可能再次阶段被用户的代表及时提醒。

● 集思广益。评审过程吸收有关的分系统人员参加,从不同侧面确认软件的协调性,从而确保软件与周围所有的技术边界符合,降低系统集成风险。

● 总结汇报。通过评审制度,使实时控制系统总指挥、总设计师了解软件生产的进度、问题和要求,做出新的部署。

由此可见开放式控制器软件的开发评审环节对于软件开发质量是十分重要的,是控制器软件质量保证体系的重要组成部分。

5 结论

软件已经成为控制器的重要组成部分。特别是开放式控制器,其开放化特质主要从其开放化软件系统中得到体现。因此,软件的质量(包括稳定性和性能)直接关系到系统的可靠性。在设计开放式控制器软件系统的工程中就要考虑到软件的测试和维护,首先从体系结构上保证系统各模块接口的清晰。此外,在开发过程中充分利用工具,及早地杜绝错误;在开发的各个环节配合相应的测试和评审。在上述基于软件工程化管理的方法的软件质量保证体系下,我们才有可能高质量地完成开放式控制器软件的开发。

参考文献:

- [1] GB/T18758.1,机械电气设备开放式数控系统第1部分[S].
- [2] 李佳特.现代CNC发展趋势[J].制造技术与机床,2003(4).
- [3] 卞立乾.关于我国数控技术发展战略初探[J].中国机械工程,2001(4).
- [4] 陈虎.开放式控制器能否成为中国装备控制器技术的新机遇[J].机械工人,2004(9):29-31.
- [5] (美) Roger S P,梅宏.软件工程实践者的研究方法[M].机械工业出版社,2002.

【上接第20页】

- Technol, 2000(16):376 - 381.
- [15] LIU Zhan-qiang. Repetitive measurement and compensation to improve workpiece machining accuracy[J]. Int J Adv Manuf Technol, 1999(15):85 - 89.
 - [16] KIM S H, KO T J, AHN J H. Elimination of settling error due to clamping forces in on-machine measurement[J]. Int J

Adv Manuf Technol, 2002, 19: 573 - 578.

- [17] AHN K G, CHO D W. In-process modelling and estimation of thermally induced errors of a machine tool during cutting [J]. Int J Adv Manuf Technol, 1999, 15: 299 - 304.
- [18] 盛伯浩,唐华.数控机床误差的动态补偿技术[J].制造技术与机床, 1997, 6.