第5卷 第4期 光学 精密工程 Vol. 5, No. 4
1997年8月 OPTICS AND PRECISION ENGINEERING August, 1997

# 实时分布式系统软件可靠性设计的一些方法

姜 韬 白超光
(中国科学院长春光学精密机械研究所, 长春130022)

摘要 介绍了软件可靠性及生存期的概念, 结合具体工程实例指出了在软件设计阶段如何通过避错设计、查错设计、改错设计、容错设计以提高软件可靠性的一些方法。
关键词: 软件可靠性; 容错设计; 避错设计

## 1 引 言

可靠性工程学迄今已有了三十多年的历史, 随着计算机科学突飞猛进的发展, 软件可靠性得到了国际国内日益广泛的重视。

计算机软件在多学科, 多领域发挥巨大作用的同时, 不可靠软件的巨大危害性也表现得更明显。尤其是对实时分布式系统, 由于软件不可靠, 会引起系统将错误的命令和状态送往各分系统, 从而使设备产生误动作甚至失效, 造成巨大的经济损失, 甚至危及人身安全。可见软件可靠性是不容忽视的大问题。下面我们将给出软件可靠性的一些基本概念, 并结合工程上实例谈谈这类软件可靠性设计的一些方法。

## 2 可靠性软件设计的一般过程

### 2.1 软件可靠性概念

(1) 在规定的条件下, 在规定的时间内, 软件不引起系统失效的概率, 该概率是系统输入和系统使用的函数 。系统输入将确定是否会遇到已存在的错误(如果错误存在的话)。

(2) 在规定的时间周期内, 在所述条件下程序执行所要求功能的能力。

收稿日期: 1997年4月3日

其中(1)是定量定义,(2)是定性定义。这个定义是由美国IEEE计算机学会认同,我国在国标GB/T-11457采用的标准。

**2.2** 应用软件工程理论设计可靠性软件

软件是一种产品,要提高软件可靠性,最根本的原则是应用软件工程理论,系统化设计软件。过去纯粹凭个人经验和技巧的作坊式编程方法是不可取的。取而代之应使用软件工程中被证明是行之有效的工具(如数据流图、HIPO图,JACKSON结构设计方法等)。软件工程理论特别强调使用软件生存周期方法学和各种结构分析和结构设计技术来设计可靠性软件。

**2.3** 软件生存期的划分

软件生存期可分为三个阶段:

- 软件定义时期
  - (a) 问题定义
  - (b) 可行性研究
  - (c) 需求分析
- 软件开发时期
  - (a) 设计阶段
  - (b) 编码阶段
  - (c) 测试阶段
- 软件维护时期

软件生存期各阶段对可靠性作用不同,图1我们给出改正一个问题需要付出的代价曲线:

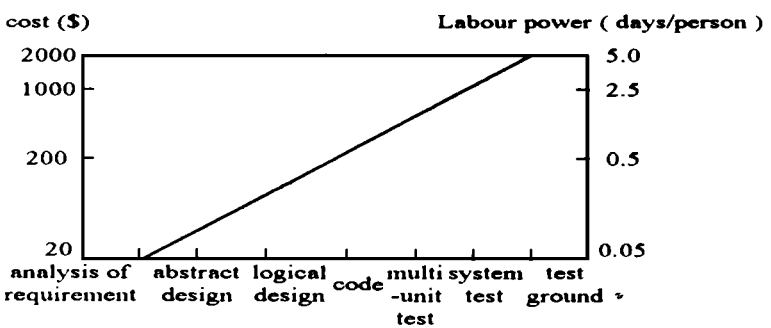


Fig. 1 Cost of correcting an error

可见完成一个可靠性软件就要在前期花费大气力。下面我们结合工程实例重点介绍一下在需求分析和设计阶段如何提高软件可靠性的一些方法。

# 3　软件可靠性设计的一些方法

## 3.1　问题定义

用户要求我们设计一套实时分布式系统,实时地显示作战态势并记录传感器传来的原始数据和中央机发出的作战命令。

## 3.2　可行性研究

该系统的主要矛盾是实时性,要求帧频20 Hz,在50 ms 内要完成显示、记录、串行通讯、并行通讯、数据处理等任务。经过仔细研究,在现有的技术条件下是可行的。

## 3.3　需求分析

软件设计始于需求分析,这部分的微小错误(即设计人员对用户理解的微小偏差)将导致不可估量的损失。因此在这一阶段要求设计人员与用户保持密切的联系,一方面设计人员应深入用户环境,另一方面主动邀请用户参与开发中的决策。

首先来看显示单元。在需求分析阶段,我们多次同用户座谈,了解他们在终端想实时看到的信息内容,这些信息的主次程度及精确程度,并主动建议用户用数字表格来显示精确的数据,对连续反映作战态势的变化的数据相关处理后以图形方式直观地显示出来。我们还设计出侧重点不同的多种界面,让用户根据使用方便作出选择和修改,最终达成一致意见:多画面切换,并根据目标距离,5级变倍显示监视区域。在这一阶段,作为设计人员,我们深入考虑了用户将来可能有的潜在要求,主动提出一些改进意见,一方面让用户满意,另一方面使软件维护方便,并保证系统完成后具有较高的先进水平。

对于记录单元,在需求分析阶段,我们要向用户了解记录的内容(向外发出的命令及传感器送来的原始数据)记录的特点(每帧记录定长还是变长,数据是帧帧有还是不定期有)及记录的介质(硬盘还是内存,如果是内存是以 EMS, XMS 或虚盘方式访问——实时记录 I/O 访问时间要求严格)。还要知道用户要求连续记录的时间(用以确定扩充内存的大小)以及记录空间满的处理方法(停止记录,或是以 FIFO 方式保留近期数据)。在倾听了用户的要求后设计人员要以技术的可行性,市场的产品行情(是否有满足要求的产品及其性能价格比),和准确的实验结果为依据提出自己的意见,以设计出既符合要求又能满足设计经费支持力度的系统。

另外,对可靠性要求严格的系统,需求分析阶段,用户和设计人员还要共同制订可靠性指标,由于各软件产品不同,并没有一个通用的标准。一般包括平均无故障时间(MTBF)、查错、改错、容错的功能,硬件失效时必要的检测和恢复能力。

## 3.4　设计阶段

结束需求分析,就进入正式的设计阶段。软件可靠性的全部内容可归结为四个类型:即避错设计、查错设计、改错设计和容错设计。

### 3.4.1　避错设计

避错设计是使软件产品在设计过程中不发生错误或少发生错误的一种设计方法。总的设计原则是控制和减少程序的复杂性。具体的方法:

(a) 系统的整体结构设计,以层次结构为最合理的结构,这可以很大程度地减少程序的复杂性。

(b) 模块化程序设计思想—— 每一个层次结构由若干模块组成。模块划分合适, 可减少开发工作量, 降低程序复杂性。原因是 C(A+ B) > C(A)+ C(B) 其中 C: 复杂度。复杂性降低意味着可靠性的提高。

运用模块化技术, 既可以将错误局限在各个模块内部, 防止错误漫延, 又可以利用以前被证明可靠的模块来构造新的系统, 减少新系统开发的工作量, 提高系统的可靠性。模块设计时以模块内聚大, 块间耦合少为好—— 模块内部各部分联系紧密, 块间联系松散, 可靠性增高。

图2我们给出上例显示单元的系统层次及模块间的关系

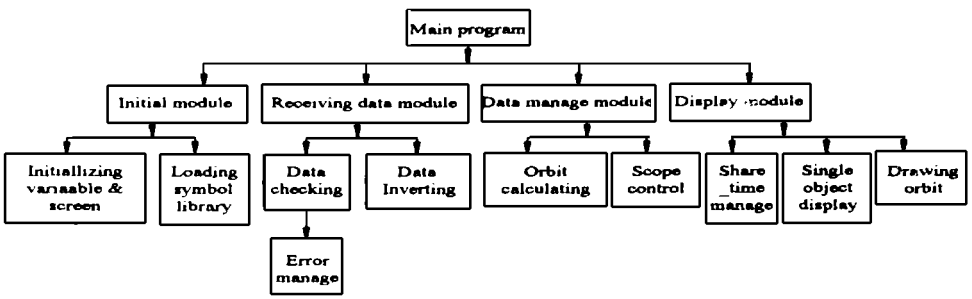


Fig. 2 Frame of the display unit

将系统结构层次化, 设计人员才能有一个清晰的思路, 便于设计和修改。模块划分时, 在每个模块能独立完成一种功能前提下, 应使模块尽可能紧凑, 块间耦合越少越好, 少到可以仅靠模块间的参数传递, 这也是可靠性系统最常使用的方式。

**3.4.2** 查错设计

在软件设计中, 正确地采用各种避错设计方法, 可以大幅度地降低引入错误, 但错误并不能完全避免, 因而查错设计(使软件产品自动纠错) 是需要的。

(a) 查错设计分为两种类型: 主动查错和被动查错。主动查错是指主动出击对程序状态的检查, 被动查错是在程序不同位置设置检测点等待错误征兆出现, 这是当前主流的检测方法。

(b) 被动检测的原则一为相互怀疑, 二为立即检测。即假定其它单元传来的数据有错, 并尽力证实。而且一有错误, 尽力证明, 降低排错难度。

(c) 被动检测的正负影响。

正面当然是利于查错, 提高可靠性。反面它使软件增加了冗余(即指错误自动检测模块), 由于它与过程处理模块串联必将降低系统的可靠性。

(d) 自动查错可用来检查数据的有效性, 尤其是输入数据(终端, 串、并口) 的有效性。也可用于检查数 据的合理性。具体包括数据属性, 上下限区间等。

在上例显示单元中的接收数据模块就调用了数据检测模块——检查输入数据的有效性计算目标航迹时, 用高低角反映目标高度, 显示作战态势。这种方法有时是不准确的, 就需用自动

查错来检测, 以调整参数来改错。

**3.4.3**　改错设计

系统一旦查出错误, 自然希望能改正错误, 改正错误的前提一是能准确地错误定位, 二是程序有能力修改错误语句。但现阶段没有人的参与几乎不可能, 最多能做的是减少损失, 限制错误的影响范围。通常采用的办法是隔离用户程序以减小失效范围, 提高可靠性。目前, 改错设计还没有达到实用阶段, 一般系统采用的是容错设计。

**3.4.4**　容错设计

软件的容错设计包括 N 文本法和恢复块法。由于大量增加的冗余部分扩大了程序的规模, 因此只用于失效后果严重的环节。

(a) N 文本法

N 文本法是指对于一个规定的功能, 由 $N(N >= 2)$ 个不同的设计组编制出 $N$ 个不同的程序, 由表决器最终完 成结果输出。它的缺点是需要 N 个机器同时工作及支撑环境。

(b) 恢复块法

恢复块容错软件的结构是

ENSURE　　T;

BY　　P;

ELSE　BY　　Q。

T: 接收条件, 是 P 或 Q 在成功执行时必须满足的标准, P 的结果不符合要求时改执行 Q。

恢复块的关键是接收检测设计, 如果接收检测不能准确地查出程序的失效, 替补过程将没有意义。替补过程根据需要可以设计多个, 依次替补。上例显示系统目标飞行高度的推算及飞行姿态的判断就采用了恢复块法。原因是各探测器 在不同环境及探测目标性质不同的情况下, 其探测能力及可信度不同, 就需要引用不同传感器数据来处理目标信息以得到可信的结果。

# 4　结　束　语

随着计算机技术的发展软件起的作用越来越大, 越来越多的控制已经由人转变为计算机软、硬件系统。软、硬件的 可靠性已经成为全球性的重大课题, 我国在大型项目尤其是军工项目对软件可靠性有了明确的要求, 但其发展水平与硬件可靠性有很大差距, 需要广大设计人员不断开发、研究。设计可靠性软件, 测试、分析、评价可靠性软件都具有深远的意义。

本文只是结合工程经验及一些软件可靠性理论探索总结而成, 所从事的也只是将这套理论同实际相结合的应用工作, 目前只是探索, 还有待于进行更多、更深入的努力。

## 参　考　文　献

[1] 袁由光等. 容错与避错技术及其应用. 北京: 科学出版社, 1992

[2] 马锦林. 软件工程引论. 南京: 南京大学出版社, 1987

# Methods on Reliable Design of Software in Real-time Distributed System

Jiang Tao , Bai Chaoguang
( *Changchun Institute of Optics and Fine Mechanics* ,
*Chinese Academy of Sciences*, *Changchun* 130022)

## Abstract

In this paper, we talk out the fundamental concepts and characteristics on reliability of software. Also, we give an example to tell you how to improve software reliability by avoiding error design, checking error design, correcting error design and fault-tolerant design in all designing stages of software .

**Keywords** : Reliability of software , Checking error design , Correcting error design , Fault-tolerant design , Avoiding-error design

姜　韬　女, 1969年11月生, 1992年毕业于大连理工大学计算机软件专业。现中科院长春光机所硕士研究生。