

## 基于控制系统负反馈框图的质量管理模型

王 典, 钟 勇, 朱政华

(中国科学院 成都计算机应用研究所, 四川 成都 610041)

(wangdian04@mails.gucas.ac.cn)

**摘 要:**提出一种使用闭环系统负反馈框图建立软件过程的控制模型的方法。该模型利用软件过程中关键域的度量值及其在不同开发阶段的权重来形式化地描述软件开发中某一时刻的特征,并且通过它与设定的门限值的比较来指导软件过程,保证软件质量。

**关键词:**软件质量;软件过程;度量模型;自动控制;负反馈

**中图分类号:**TP311.5 **文献标识码:**A

## Software management model based on negative feedback model

WANG Dian, ZHONG Yong, ZHU Zheng-hua

(Chengdu Institute of Computer Application, Chinese Academy of Sciences, Chengdu Sichuan 610041, China)

**Abstract:** A method for modeling software process using close loop control system analysis was proposed. The model formalized the characteristics of every different phases in developing process according to key practice-area measurement value and its weight. Compared with the threshold value set before, it guided the process of software development, and ensured software quality.

**Key words:** software quality; software process; measure model; automation; negative feedback

## 0 引言

在当前的软件管理过程中,对于项目的阶段性评价多是由专家以表格的形式来描述的。传统的度量方法如 GQM, GDSM 等都是将组织的商业目标逐级分解从而得出度量的标准,它们给出了建立度量的基本步骤,但是缺乏形式化的支持。

把自动控制原理引入软件开发是 Abdelzaher 等人<sup>[1]</sup>首先提出的,而后一些文章<sup>[2,3]</sup>对其在分析软件运行过程中的稳定性和鲁棒性方面进行了进一步的研究。建立控制系统模型的好处在于可以通过系统的时域及频域分析,利用现有的结论如劳斯稳定性判据等分析系统性能。

要利用控制领域的结论首先要建立系统的闭环模型。控制系统模型由输入矩阵  $r(t)$ , 输出矩阵  $X(t)$  和传递参数  $A$  等组成。本文将控制的概念引入开发过程中,提出了建立软件开发过程的控制系统闭环模型,确定系统参数的方法,并通过此模型提出了一种泛用的软件过程控制方法。

## 1 软件开发的控制系统模型

### 1.1 闭环系统

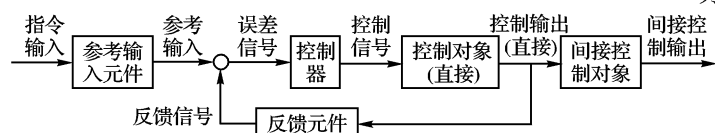


图1 闭环系统组成框图

闭环系统为自动控制中常用的概念。这种系统的输出信号要引回输入端以影响对系统的控制,所以它的信号流通构成了闭合回路。这种输出信号对控制作用有直接影响的系

统,称之为闭环系统(图1)<sup>[4]</sup>。

闭环系统也就是反馈系统,反馈信号可以是输出信号本身,也可以是输出量的函数。对输入信号和反馈信号进行比较就是求二者之差,因此,这种反馈又被称为负反馈(negative feedback)。

### 1.2 软件开发的控制系统模型

软件生命周期的每一个阶段均可以使用闭环系统模型来描述。以建模阶段为例。

按照经典的软件生命周期模型,软件开发的第一个阶段是建模阶段。软件建模阶段的输入是建立模型书写文档等工作,输出是软件模型。输出的软件模型可以用一组反映其特征的变量来描述<sup>[2]</sup>。考虑到我们的输入和输出并非如控制电路一样是标准单变量输入输出(即并非指令输入输出,而是一个描述性的向量组),对于使用到的变量作如下规定: $r(t)$  为输入信号(向量),指客户需求的模型的特征向量(为理想情况,对于一个确定的输入量,相当于一个阶跃信号); $u(t)$  为误差信号(向量),指剩余工作和工作情况的特征向量; $X(t)$  为输出信号(向量),指  $t$  时刻的工作成果特征向量; $A$  为系统控制矩阵,相当于典型闭环控制系统框图中的控制器; $K$  为状态反馈矩阵,相当于典型闭环控制系统框图中的反馈元件。

## 2 系统建模阶段的度量

### 2.1 传递函数的确定

在软件开发过程中,受到开发人员和开发条件的限制,对输入的误差信号  $u(t)$  的控制结果是开发软件的效率(或者说是表示软件开发速度的量),将其特征向量用  $n(t)$  来

收稿日期:2006-04-30

**作者简介:**王典(1982-),男,北京人,硕士研究生,主要研究方向:软件过程、软件质量;钟勇(1966-),男,四川岳池人,研究员,博士生导师,博士,主要研究方向:软件过程、软件质量、现代集成制造系统;朱政华(1979-),男,湖南郴州人,硕士研究生,主要研究方向:软件过程、软件质量。

表示。 $u(t)$  和  $n(t)$  之间的转换函数设为  $B$ 。有:

$$n(t) = u(t) * B \quad (1)$$

作为输出结果的  $X(t)$  即  $t$  时刻工作完成情况的特征向量为工作效率  $n(t)$  对  $t$  的积分。即:

$$u(t) * A = B * \int u(t) dt \quad (2)$$

为了确定传递函数  $B$ , 首先要明确定义  $X(t)$  和  $u(t)$ 。在文献[5]中应用 GQM (Goal-Question-Metric) 模型对需求中关

键过程域进行度量。提出了三个度量目标:1) 跟踪软件需求的状态和质量; 2) 控制需求的变更; 3) 跟踪需求处理的速度。并对每一个目标定义了一组度量。经过整理, 取其能够描述当前状态的一组特征如表 1 所示。

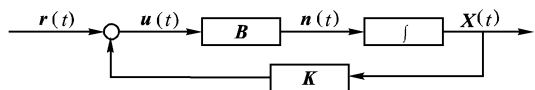


图 2 系统建模阶段的闭环系统模型

表 1 需求管理关键域度量

需求管理关键域	问题	度量
处于“批准的”状态的需求数的度量	需求的评审结果怎样?	参加评审的需求占总需求的比例
		经评审认为不可行的无法实现的需求率 *
		认为说明不清楚、不适当的需求率 *
		认为与其他需求不一致的需求率 *
控制需求的变更	评审认为有问题的需求是否已修改	认为无法测试的需求率 *
		评审认为有问题的需求率 *
	提出了多少次需求的变更?	经过了修改并被接受的需求率 *
		提出的需求变更数目占总需求数之比 *
		未决定的需求变更率 *
		认可的需求变更率 *
	需求变更的类型	纳入系统基线的需求变更率 *
		放弃的需求率 *
		暂时删除, 在以后的版本中实现的需求率 *
		加入的新的需求个数占总需求之比 *
跟踪需求处理的速度	需求变更的影响	引起现有约定变化的需求变更个数占总需求之比 *
		引起软件计划变化的需求变更个数占总需求之比 *
		引起工作产品发生变化的需求变更个数占总需求之比 *
		需求的处理速度是否有所改善? (现在的平均处理时间/平均处理时间的基线) 100% *
	需求的处理速度是否达到了期望值?	(现在的平均处理时间/期望的处理时间) 100% *
		需求变更的处理速度是否有所改善? (现在的平均变更处理时间/平均变更处理时间的基线) 100% *
		需求变更的处理速度是否达到了期望值? (现在的平均变更处理时间/期望的变更处理时间) 100% *

$X(t)$  为上面定义的度量标准的值所组成的向量, 它描述了当前状态下需求模型的特征。与此类似的  $n(t)$  为上面定义的度量标准的微分, 即单位时间内的各项度量。

$$X(t) = (x_1, x_2, x_3, \dots, x_n) \quad (3)$$

$$n(t) = (n_1, n_2, n_3, \dots, n_n) \quad (4)$$

由此我们得到了系统的微分方程为:

$$X(t) = \int (r(t) - X(t) \cdot K) \cdot B dt \quad (5)$$

整理后得到:

$$\dot{X}(t) = -X(t) \cdot K \cdot B + r(t) \cdot B \quad (5)$$

称之为系统的状态空间模型。

在软件开发的需求建模阶段, 在时间  $t$  时审核的是当前系统的建模情况。因此状态反馈矩阵  $K$  为单位矩阵。

而由于软件开发过程是一个非线性系统, 即  $n(t)$  不是随着  $u(t)$  线性变化的。因此, 要得到  $B$ , 还需要经过大量的实验。而后通过系统辨识等方面的相关知识, 推导出  $B$  来。依照目前的模型, 确定了  $B$  之后, 对于一个确定的输入, 应该有一个确定的输出。但是不同的建模人员, 建立模型的质量差别有时相当大, 因此这个闭环系统存在扰动。

## 2.2 扰动

在理解需求的过程中, 不可避免地会有理解的偏差, 这就造成了需求建模的不准确。在控制系统里, 将之归为扰动。扰

动是加于系统的不希望有的作用, 它对控制输出结果有不利影响。

在软件开发建模阶段的模型中, 在将误差信号  $u(t)$  转化为建模成果  $n(t)$  时, 无意识地加入了人为的干扰, 即信号干扰是加在  $B$  上的。设扰动为  $d(t)$ 。 $d(t)$  是对软件开发者的重要的评价标准之一。

在需求管理的关键域度量中, 有很大一部分度量标准是和系统扰动相关的 (参见表 1 带 \* 的度量)。它们既反映了当前工作情况的特征, 又反映了软件开发者的情况即扰动的大小。

系统扰动作用的另一个表征是系统缺陷。在文献[8]中提到了如何利用发现的系统缺陷分析软件质量。

添加了扰动并且设  $K = 1$  时的闭环系统如图 3 所示。

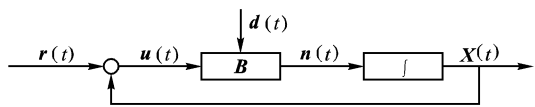


图 3 添加了扰动的闭环系统模型

当前状态下的  $d(t)$  是不可知的, 能够得到的是前一时刻的扰动值。通过对历史数据的分析, 可以大致了解到开发小组的工作情况 (这一点可以参考能力成熟度模型 CMM 和 CMMI 中关于建模方面的描述)。同样根据需求管理相关的关键域的度量来衡量。

### 3 利用闭环系统模型指导开发过程

建立软件开发的闭环控制系统模型是为了指导软件过程,保障软件质量。在需求建模阶段,需要根据  $u(t)$  来决定需求建模的进程。当  $u(t)$  达到一定标准时,进入软件开发的下一阶段或者继续进行此阶段,否则暂停当前环节,回溯到上一环节以调节此环节输入。

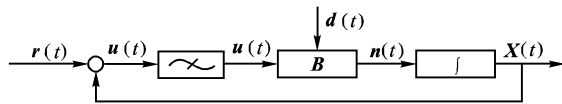


图4 添加了滤波器的闭环系统模型

为此需变换  $u(t)$ 。在文献[6]中提到了在软件开发的各阶段,对于不同方面要求的权重分配,对软件开发各阶段提出的参考权重如表3。

表2 各阶段需求权重

$C$	$SubC$	$Pl$	$De$	$Co$	$Te$	$Ma$
$F$	Accuracy	0.038	0.045	0.063	0.045	0.029
	Interoperability	0.044	0.040	0.055	0.046	0.034
	Security	0.037	0.39	0.041	0.035	0.031
	Compliance	0.050	0.040	0.052	0.039	0.032
$R$	Maturity	0.078	0.055	0.056	0.054	0.052
	Fault tolerance	0.068	0.071	0.058	0.064	0.053
	Recoverability	0.054	0.067	0.054	0.079	0.054
$U$	Understandability	0.050	0.040	0.035	0.039	0.031
	Learn ability	0.044	0.041	0.033	0.037	0.034
	Operability	0.047	0.050	0.046	0.061	0.040
	Attractiveness	0.047	0.043	0.037	0.045	0.035
$E$	Time behavior	0.090	0.080	0.072	0.078	0.074
	Resource-utilization	0.077	0.088	0.093	0.078	0.086
$M$	Analyzability	0.035	0.034	0.026	0.028	0.051
	Changeability	0.031	0.038	0.033	0.036	0.065
	Stability	0.035	0.047	0.049	0.048	0.063
	Testability	0.035	0.039	0.048	0.046	0.059
$P$	Adaptability	0.037	0.037	0.037	0.037	0.043
	Install ability	0.029	0.030	0.038	0.034	0.040
	Co-existence	0.039	0.039	0.038	0.039	0.051
	Replace ability	0.035	0.037	0.036	0.036	0.043

同时考虑到需求管理的关键域,取设计列的权值与其相应的特征变量相乘并且累加得到当前产品的评价值。

设特征变量关系矩阵为  $C$ , 权值与输出相对应的矩阵为  $W$ , 当前时刻的评价值为  $s(t)$ , 有:

$$s(t) = X(t) * C * W \quad (7)$$

其中  $X(t)$  为  $1 * n$  矩阵,  $C$  为  $n * n$  矩阵,  $W$  为  $n * 1$  矩阵。

根据 McCabe 走查方法提出三要素思想:复杂度的选择、门限值选择和超标值的识别。本文规定了一种控制原则:1) 根据项目复杂度分别设置几组门限值,作为不同规模的系统在不同要求下的门限值;2) 向量数值化(即计算出  $s(t)$ );3) 根据选择的门限值比对进行控制。

再通过领域专家的评价,我们可以得到一般情况下不同规模、不同质量要求的开发模型的门限值,并由选择的门限值和计算的  $s(t)$  来控制软件开发的过程。

反映到闭环系统的模型上,就相当于添加了一个滤波器,规定当高/低于选择的门限值时,应当停止当前阶段工作,回

溯到上一阶段完善。

### 4 实例与验证

在一个小型项目的开发实验中,得到某次评审的结果参照表1的判断标准整理如下:

$$X(t) = \begin{bmatrix} \frac{37}{53} & \frac{3}{37} & 0 & \frac{2}{37} & 0 & \frac{5}{37} & \frac{5}{37} & \frac{5}{53} & \frac{1}{53} & \frac{4}{53} & 0 & 0 & \frac{4}{53} & \frac{12}{53} & 0 & 0 & 0 \\ \frac{1}{5} & \frac{24}{53} & \frac{24}{53} & 1 & \frac{1}{5} & \frac{1}{24} & \frac{53}{24} & 1 & \frac{1}{5} & \frac{1}{24} & \frac{53}{24} & 1 & \frac{1}{5} & \frac{1}{24} & \frac{53}{24} & 1 & \frac{1}{5} \end{bmatrix}$$

为了说明方便,在此取  $C$  为单位矩阵,调整  $W$  的取值求  $s(t)$ 。在实际应用中,  $W$  为确定值,通过  $C$  的取值表现特征与权值关系。

$$W^* = [0.055 \quad -0.045 \quad -0.039 \quad -0.039 \quad -0.050 \quad 0.067 \quad -0.055 \quad 0.067 \quad -0.035 \quad 0.035 \quad -0.055 \quad -0.042 \quad -0.067 \quad 0 \quad -0.055 \quad -0.067 \quad -0.067 \quad 0 \quad -0.080 \quad 0 \quad -0.080]^T$$

经计算得到  $s(t) = -0.026$ , 即实验项目在此时刻的评价值。

### 5 结语

本文提出了一种使用控制系统负反馈框图定量描述软件开发过程的方法,建立了软件开发的控制系统模型,并且给出了软件过程控制的原则。通过开发阶段某一时刻的特征来判断整个开发过程是否健康,从而避免了无效的工作,进而保障软件的质量。

由于篇幅所限,本文仅以需求建模阶段为例阐明观点。并且只是给出了形式化的方法,并没有给出特定阶段具体的控制方程。在最后的实证工作中也是使用根据表2估计的权值矩阵替代了变换矩阵  $C$ 。在下一步的工作中,应该通过对一些具体的开发项目的评审,确定传递函数  $B$  和不同条件下作为控制评价标准的门限值。同时,以今后的工作中,也应将控制模型扩展到软件开发的其他阶段。

#### 参考文献:

- [1] ABDELZAHER TF, STANKOVIC JA, LU C, et al. Feedback performance control in software services [J]. IEEE Control Systems Magazine, 2003, 23(3): 74-90.
- [2] AMIRIJOO M, HANSSON J. Robust Quality Management for Differentiated Imprecise Data Services [A]. Proceedings of the 25th IEEE Real-Time Systems Symposium (RTSS 2004) [C]. IEEE Computer Society, 2004.
- [3] AMIRIJOO M, HANSSON J, GUNNARSSON S. Enhancing Feedback Control Scheduling Performance by On-line Quantification and Suppression of Measurement Disturbance [A]. Proceedings of the 11th IEEE International Real-time and Embedded Technology and Applications Symposium (RTAS 2005) [C], 2005.
- [4] 杨位钦 谢锡祺. 自动控制理论基础 上册[M]. 北京: 北京理工大学出版社, 1991.
- [5] 朱丽娟 钱红兵. 需求管理关键过程域的度量[J]. 计算机工程与应用, 2003, 40(24): 64.
- [6] LEE K, LEE SY. A Quantitative Software Quality Evaluation Model for the Artifacts of Component Based Development [A]. SNPD/SAWN 2005 [C]. IEEE, 2005. 20.