

基于架构风格的软件可靠性评估^①

Software Reliability Evaluation Based on Architectural Styles

黄 宁 陈未如 (沈阳化工学院 计算机科学与技术学院 辽宁 沈阳 110142)

摘 要: 基于架构的软件可靠性评估是近年来软件工程领域的研究热点之一。从软件架构风格角度开展对软件架构可靠性评估方法的研究可以在软件开发初期为设计人员提供架构选取的依据。研究了不同架构风格的特点,从组件拓扑结构角度建立了线型、星型风格的可靠性评估模型。在研究 4 种基础架构可靠性计算公式的基础上,对这两种模型的可靠性评估方法分别进行了讨论。

关键词: 软件架构 架构风格 基础架构 组件 可靠性评

软件架构是软件系统中的核心元素,它直接决定了软件系统的运行框架^[1]。对于一个软件系统,虽然好的架构并不一定能保证可以获得高可靠性的软件产品,但可以肯定的是,建立在不良架构基础上的软件不可能获得较高的可靠性^[2]。从架构入手,通过结构改善,不仅可以控制问题的复杂性,也是保证软件产品质量,减少软件开发费用的一个重要方法。为此,一些学者建立了基于组件的可靠性评估模型,用于在软件设计初期预测软件的可靠性。这些模型大多是研究基于马尔科夫链的软件组件与软件可靠性的关系,很少考虑架构风格对可靠性的影响。而从架构风格角度开展对软件架构可靠性评估方法的研究,不仅有利于简化分析过程,实现自顶向下的逐层分析,提出标准评估方法,而且为设计人员提供了架构选取的依据。

从组件拓扑结构角度,架构风格可分为 4 类:线型结构风格、网状结构风格、星型结构风格、总线结构风格。本文介绍了 4 种基础架构可靠性计算方法,并研究了线型、星型风格的可靠性评估模型。

1 基础架构可靠性计算方法

软件系统是一个逻辑的整体,大到整个系统、子系统,小到每个组件甚至组件中的每个接口实现,它们之间都有着一定的联系,具有结构特性。在基于组件的软件架构中,组件是最基本的计算单元,它们之间的基本结构关系包括:冗余架构和分支架构。理论

上,一个软件系统总可以分解成冗余架构、分支架构的一种或它们的组合。若知道这两种架构的可靠性计算方法,进而就可以计算整个系统的可靠性。

下面说明 4 种架构的可靠性计算方法。这里假设:互不包含组件之间相互独立,任意一个组件的可靠性都不受任何其它组件可靠性的影响。

1.1 冗余架构

定义 1. 设系统由 n 个组件组成,如果当且仅当 n 个组件全部失效时,系统才失效,任意一个组件正常工作都可保证系统正常工作^[3],这样的结构称为冗余架构。架构关系如图 1 所示。

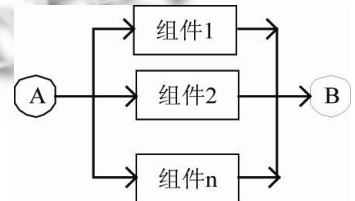


图 1 组件间的冗余架构

设第 i 个组件的可靠性为 R_i , 则冗余架构的可靠性计算公式为:

$$R = 1 - \prod_{i=1}^n (1 - R_i) \quad (1)$$

证明: 设第 i 个子系统(组件)的寿命为 X_i , 各子系统相互独立,在时间 t 时的可靠度为 $R_i(t)$, 系统寿命为

① 收稿时间:2008-10-21

X_s , 其在时间 t 时的可靠度为 $R(t)$, 则:

$$\begin{aligned}
 R(t) &= P\{X_s > t\} = P\{\max(X_1, X_2, \dots, X_n) > t\} \\
 &= 1 - P\{X_1 \leq t, X_2 \leq t, \dots, X_n \leq t\} \\
 &= 1 - \prod_{i=1}^n P\{X_i \leq t\} \\
 &= 1 - \prod_{i=1}^n (1 - R_i(t))
 \end{aligned}$$

1.2 串联架构

定义 2. 设系统由 n 个组件组成, 如果当且仅当 n 个组件全部正常工作时, 系统才正常工作, 任意一个组件失效都将导致系统失效^[3], 这样的结构称为串联架构。架构关系如图 2 所示。



图 2 组件间的串联架构

设第 i 个组件的可靠性为 R_i , 则串联架构的可靠性计算公式为:

$$R = \prod_{i=1}^n R_i \quad (2)$$

证明: 设第 i 个子系统(组件)的寿命为 X_i , 各子系统相互独立, 在时间 t 时的可靠度为 $R_i(t)$, 系统寿命为 X_s , 其在时间 t 时的可靠度为 $R(t)$, 则:

$$\begin{aligned}
 R(t) &= P\{X_s > t\} = P\{\min(X_1, X_2, \dots, X_n) > t\} \\
 &= P\{X_1 > t, X_2 > t, \dots, X_n > t\} \\
 &= \prod_{i=1}^n P\{X_i > t\} \\
 &= \prod_{i=1}^n R_i(t)
 \end{aligned}$$

1.3 分支架构

定义 3. 设系统由 n 个分支组成, 每个分支对应系统的一个操作剖面, 某一个分支的失效, 将以相应的概率影响系统的可靠性, 这样的结构称为分支架构。架构关系如图 3 所示。

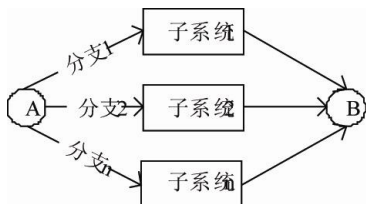


图 3 组件间的分支架构

设第 i 个分支的可靠性为 R_i , 操作流经过第 i 个分支的概率为 P_i , 则分支架构的可靠性计算公式为:

$$R = \sum_{i=1}^n (P_i (1 - P_i (1 - R_i))) \quad (3)$$

串联架构是分支架构的一个特例。数据流将以 100% 的概率流向每个组件, 即 $P_i=1$ 。代入公式(3), 可得到串联可靠性计算公式:

$$R = \prod_{i=1}^n R_i$$

1.4 循环架构

定义 4. 设系统由一个循环 m 次的组件(子系统)组成。如果当且仅当组件正常循环 m 次后, 系统才正常工作, 任意一次循环失效都将导致系统失效, 这样的结构称为循环架构。架构关系如图 4 所示。

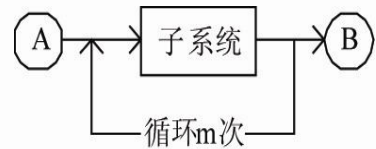


图 4 组件间的循环架构

设组件/子系统的可靠性为 r , 则循环架构的可靠性计算公式为:

$$R = r^m \quad (4)$$

证明: 循环架构可以看成是由 m 个子系统或组件组成的串联架构, 由公式 2 知 $R = \prod_{i=1}^m r = r^m$ 。

2 连接件与架构风格

连接件的形式取决于软件架构风格, 不同软件架构风格的连接件形式也不相同, 如管道/过滤器风格中的管道, C2 风格中的消息路由装置等, 不同连接件实现了不同连接语义^[4]。表 1 从拓扑结构角度对软件架构风格进行了分类并给出了风格与连接件的对应关系。

3 不同软件架构风格可靠性计算方法

拓扑结构为线型结构的软件架构, 系统从整体上可以看成是由多个组件和连接件连接的串联架构。因此, 对这类架构风格我们总体上采用串联架构进行可靠性计算。线型结构典型的例子有: 分层风格, 管道过滤器风格等。

表 1 连接件与软件架构风格的对应关系

风格分类	风格	主要连接机制
线性结构	成批处理	数据流
	管道过滤器	数据流 / 过程调用
	分层	过程调用
	正交	过程调用
网状结构	面向对象	过程调用
星型结构	黑板、仓库	共享数据访问
	客户服务器	过程调用
	浏览器服务器	过程调用
总线结构	基于事件	事件驱动
	消息总线	消息传递

3.1 分层系统

层次式架构风格是把大型软件系统按照功能的扩展性，分成若干层，每一层向其上层提供服务，并利用下层的的服务。在一些层次系统中，除了一些精心挑选的输出函数外，内部的层只对相邻的层可见。协议定义了连接器，也决定了各层之间如何进行交互。分层风格结构框图如图 5 所示。

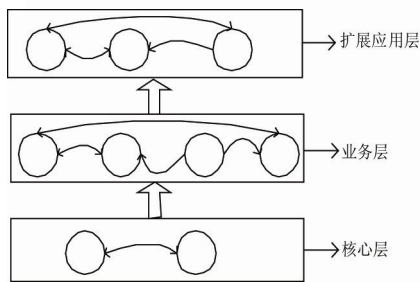


图 5 层次风格

关于层次风格的可靠性，有学者做了相关的研究，2002 年 S. P. Lebanc^[5]提出将层次风格转化成关联架构进行可靠性计算，并得出分层系统可靠性计算公式(5)，其中表示第 j 层第 i 个组件的可靠性。该公式假设：①各组件相互独立；②任何组件的失效都将直

接引起系统的失效；③各组件被执行的概率相等。依据上述假设，可知系统能够正常运行的条件是各组件均能正常运行，即各组件之间构成了一种串联架构。

$$R = \prod_{j=1}^n \prod_{i=1}^{m_j} R_{ij} \quad (5)$$

随着网络的发展，很多软件都采用层次风格作为系统的总体结构，以此适应分布式系统的广泛应用，而内部结构根据不同的功能及性质采用相应的风格。根据这一特点，在进行层次风格可靠性评估时，可将一层看作一个复合组件。层里视具体情况再进行风格分割。用 R_i 来表示各个层次的可靠性，假设任意一层的失效都导致整个系统失效，由公式(2)，则层次关系的可靠性计算公式为：

$$R_{in} = \prod R_i \quad (6)$$

层次内部之间的关系可根据组件之间的基础架构关系进行计算，而不仅仅采用公式(1)方式把组件都看作是串联的关系。

3.2 星型结构风格

拓扑结构为星形结构的软件架构，各组件之间相互独立，只能通过一个或多个中心组件进行通信，系统从整体上呈星状分布。从中心组件的角度来看，其它组件之间的关系是并行的分支结构。因此，对这类架构风格我们总体上采用分支架构进行可靠性计算。星型结构典型的例子有：黑板、仓库风格，客户服务器风格，浏览器服务器风格。

3.2.1 黑板、仓库风格

黑板、仓库的特点是有一个共享中心数据区，为相关软件提供数据访问和存储服务。根据通信方式的不同，产生两个主要的子类：黑板和仓库。黑板会主动的在知识源所关心的数据发生变化时向知识源发送消息，而仓库则只是一个被动式的数据存储单元(如一个文件)，供知识源向其中写入或读出数据。

从黑板的角度来看，各知识源之间的关系就是并行的分支结构。每个知识源对整个系统的功能的实现贡献率为 P_i 。设黑板的可靠性为 R_b ，知识源的可靠性为 R_k 。黑板、仓库风格软件架构可靠性计算公式为(客户服务器、浏览器服务器风格类似)：

$$R = R_b \times \prod_{i=1}^n (1 - P_i(1 - R_k)) \quad (7)$$

4 简单实例

图 6 是一个简化的数据库系统(仓库风格)。假设 $RA=0.996$, $RB1=RB2=0.986$, $RB3=RB4=0.988$, 根据测试数据, 数据流进入各个分支的概率为 $P1=0.02$, $P2=0.102$, $P3=0.141$, $P4=0.737$ 。

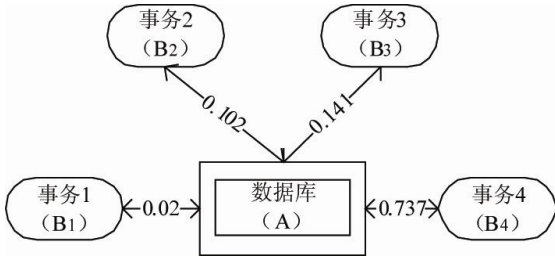


图 6 简单知识库数据架构

由仓库风格软件架构可靠性计算公式(6), 可知系统可靠性为:

$$\begin{aligned}
 R &= R_A \times \prod_{i=1}^4 (1 - P_i(1 - R_i)) \\
 &= 0.996 \times [1 - 0.02 \times (1 - 0.986)] \times [1 - 0.102 \times (1 - 0.986)] \\
 &\quad \times [1 - 0.141 \times (1 - 0.988)] \times [1 - 0.737 \times (1 - 0.988)] \\
 &= 0.984
 \end{aligned}$$

5 结束语

研究架构风格不仅有助于更准确地把握具有特定风格的软件系统的各种特征, 以便设计人员能在系统结构级上尽早达成共识, 提高系统的可重用性, 而且能帮助设计人员更清楚地了解不同架构风格间的异同

点, 为使用和组合不同风格的组件提供指导^[6]。本文给出了一种基于风格的可靠性评估方法, 将架构风格进行了拓扑结构分类, 建立了不同风格的可靠性评估模型。该模型为软件设计初期架构风格的选择提供了依据: 在统中存在多种架构风格选择的情况下, 可根据架构风格可靠性高低进行选取。同时在计算基于架构风格的软件系统可靠性时, 不需要将系统一次性划分为一个个不可分组件, 而是根据架构风格进行划分, 从而降低了可靠性计算的复杂性。

基于组件的软件开发方法是一种发展趋势, 研究基于组件软件的软件架构设计及评估方法是目前的热门课题。虽然还处于起步阶段, 相信一定会为提高软件开发效率、保证软件开发质量起到重要的作用。

参考文献

- 1 王小鹏. 基于模式的软件体系结构设计. 软件导刊, 2007(1):29-31.
- 2 周欣, 黄璜, 孙家骕. 软件体系结构质量评价概述. 计算机科学, 2003, 30(1):49-52.
- 3 马敏书, 张仲义, 吕永波. 层次型软件系统可靠性模型及预测. 中国软科学, 2003, 12(6):147-150.
- 4 刘瑜, 张世君, 王立福. 连接件模型与分类研究. 计算机工程, 2005, 2(3):77-79.
- 5 Lebac SP, Roman PA. Reliability Estimation of Hierarchical Software Systems. Proceedings Annual Reliability and Maintainability Symposium, 2002.
- 6 焦文品, 史忠植. 用 XYZ/E 形式化体系结构风格. 软件学报, 2000, 11(3):410-415.