

网构软件可靠性代数模型及简化计算研究

张 靖^{1,2}, 雷 航¹

(1 电子科技大学 计算机科学与工程学院, 四川 成都 610054; 2 攀枝花学院 网络中心, 四川 攀枝花 617000)

摘 要: 为了在网络环境下为网构软件提供一致的语法和语义规约并进行可靠性分析, 利用抽象代数理论, 根据网构软件特征, 建立了网构软件形式化代数模型, 在此基础上实现了构件组合运算, 在定性和定量方面具有演算能力, 提出了网构软件系统可靠性计算的简化方法. 实验证明, 在网构软件的形式化分析基础上建立的模型和提出的可靠性计算方法正确、有效, 可以有效地应用于网构软件设计、开发和测试、维护等方面.

关键词: 网构软件; 可靠性; 形式化; 演算; 模型; 简化方法

中图分类号: TP301.2

文献标识码: A

文章编号: 1000-7180(2013)07-0028-05

Research on the Algebra Model and the Method of Simplified Calculating of Internetwork Reliability

ZHANG Jing^{1,2}, LEI Hang¹

(1 School of Computer Science and Engineering, Univ. of Electron. Sci. & Tech. of China, Chengdu 610054, China;

2 Campus Computer Network Center, Panzhihua University, Panzhihua 617000, China)

Abstract: To provide a consistent syntax and semantics specification of Internetwork and implement the reliability analysis under Internet environment, by using the theory of Abstract Algebra, the formalization algebraic model of Internetwork is established based on the character of Internetwork, and the component composition computing of Internetwork is realized. The model has the calculus ability with the qualitative and quantitative aspects. A simplified calculating method is presented in this paper, it is proved that the model and the method, which is constructed based on formalization analysis of Internetwork, is right and available, and they will be applied to Internet design, development, testing, maintenance and so on.

Key words: internetwork; reliability; formalization; calculus; model; simplified method

1 引言

网构软件(Internetwork)作为一种新的软件模式,可以在 Internet 网络环境下实现软件资源共享、重复利用,形成统一的资源平台、计算平台、服务平台,提供优良的智能化服务. 相关的研究已经在国内外展开,文献[1]进行了网构软件系统构建的形式化分析和研究,但缺乏实际的演算能力,文献[2]对软件体系结构做了形式化分析和建型,但没有针对性的进行网构软件形式化研究,文献[3]研究了网构软件组装的可靠性及敏感度,但缺少系统整体可

靠性评估方法,文献[4]分析了网构软件的可靠模型及种类,讨论了容错和冗余技术,但没有具体的可靠性形式化表达及评估方法,文献[5]将服务更新强度引入(M-O)模型中进行可靠性分析,但可靠性计算较复杂,文献[6]提出了网构软件的演化算法和引入监测、反馈实现网构软件可靠性评估方法,方法比较复杂,也缺乏有效形式化分析建模和系统可靠性计算. 因此,利用抽象代数理论,探讨网构软件系统结构定义和形式化分析,便于为构件在发现、选择、演化等方面提供一致的语法和语义规约,研究可靠性计算的简化方法便于在网构软件可靠性设计、

收稿日期: 2012-10-05; **修回日期:** 2012-11-07

基金项目: 四川省科技项目基金项目(2009F80152)

分配以及开发、维护过程中提供量化分析,具有十分重要的价值。

2 网构软件代数模型

2.1 网构软件模型

根据网构软件特征,其模型定义如下。

定义 1 网构软件由网络上的构件组成,每一构件是一种计算单元,构件定义为一个多元组,即 $I = (ID, interf, funct, depend, descri)$ 。其中, ID 表示构件名称标识, $interf$ 表示构件接口, $funct$ 表示构件功能行为, $depend$ 表示构件之间的依赖关系, $descri$ 表示构件的说明,主要是构件运行初始条件($init$)、前置条件($pre - condi$)、后置条件($post - condi$)及软件可靠性等非功能性说明。

一般网构软件之间的连接部分可视为连接构件。

定义 2 构件接口是构件运行时与环境的输入/输出信息,即 $interf = (interfI, interfO)$,其中 $interfI$ 是构件运行所需要的环境或者其他构件功能集合, $interfO$ 是构件运行提供给环境或者其他构件功能集合。

定义 3 构件接口依赖关系是构件运行时与其他构件之间直接作用的依赖关系,即 $depend = ((\delta(id_i, \alpha, id_j), p_{ij}), \alpha \in Funct, id_i, id_j \in ID, \delta$ 是映射函数,表示在功能行为 α 下构件 1 作用构件 2,之间发生依赖关系(转移、调用等)概率是 p_{ij} 。

定义 4 构件可靠度表示网构软件在一定环境下正常运行的概率,即 $R = r_d \times l_{ij}$, r_d, l_{ij} 分别表示构件本身可靠和构件执行动作路径的可靠度,路径的可靠度可以用转移概率度量。

定义 5 设论域 $Dom(U)$, 构件集合为 $U, A, B \in U$, 如果满足下列条件,则 A, B 相等,即 $A = B$ 。

- (1) $Dom(A) = Dom(B)$;
- (2) $interfI(A) = interfI(B)$;
- (3) $interfO(A) = interfO(B)$;
- (4) $funct(A) \Leftrightarrow funct(B)$;
- (5) $depend(A) \Leftrightarrow depend(B)$;
- (6) $descri(A) \Leftrightarrow descri(B)$ 。

定义 6 设论域 $Dom(U)$, 构件集合为 $U, A, B \in U$, 如果满足下列条件,则 A 是 B 的演化,即 $E - evolve(B, A)$ 。

- (1) $Dom(A) = Dom(B)$;
- (2) $interfI(A) \supseteq interfI(B)$;
- (3) $interfO(A) \subseteq interfO(B)$;
- (4) $funct(A) \Rightarrow funct(B)$;

(5) $depend(A) \Rightarrow depend(B)$;

(6) $descri(A) \Leftrightarrow descri(B)$ or $descri(A) \Rightarrow descri(B)$ 。

2.2 网构软件组合运算

定义 7 OP 表示构件全部动作行为集合, $op_i \in OP$, 构件组合运算语法定义如下:

$OP ::= op_1 \rightarrow op_2 \mid op_1 + op_2 \mid op_1 \leftrightarrow op_2$

① $op_1 \rightarrow op_2$ 表示顺序执行,先执行 op_1 ,结束后继续执行 op_2 ;

② $op_1 + op_2$ 表示选择执行,或执行 op_1 ,或执行 op_2 ,选择是不特定的,依赖于系统具体运行环境;

③ $op_1 \leftrightarrow op_2$ 表示循序执行, op_1 和 op_2 间循环执行。

定义 8 $c_1, c_2 \in U, c_1 \rightarrow c_2$ 顺序执行运算组合成一个新构件,且

$Dom(c_1 \rightarrow c_2) = Dom(c_1) \cup Dom(c_2)$;

$interf(c_1 \rightarrow c_2) = interfI(c_1) \cup interfO(c_2)$;

$funct(c_1 \rightarrow c_2) = funct(c_1) \cup funct(c_2)$;

$depend(c_1 \rightarrow c_2) = depend(c_2)$;

$descri(c_1 \rightarrow c_2) = descri(c_1 (init, pre - condi)) \cup descri(c_2 (init, post - condi))$ 。

定义 9 $c_1, c_2 \in U, c_1 + c_2$ 选择执行运算组合成一个新构件,且

$Dom(c_1 + c_2) = Dom(c_1) \text{ or } Dom(c_2)$;

$interf(c_1 + c_2) = interfI(c_1) \text{ or } interfO(c_2)$;

$funct(c_1 + c_2) = funct(c_1) \text{ or } funct(c_2)$;

$depend(c_1 + c_2) = depend(c_1) \text{ or } depend(c_2)$;

$descri(c_1 + c_2) = descri(c_1) \text{ or } descri(c_2)$ 。

定义 10 $c_1, c_2 \in U, c_1 \leftrightarrow c_2$ 循环执行运算组合成一个新构件,且

$Dom(c_1 \leftrightarrow c_2) = Dom(c_1) \cup Dom(c_2)$;

$interf(c_1 \leftrightarrow c_2) = interfI(c_1) \cup interfO(c_1) \cup interfO(c_2)$;

$funct(c_1 \leftrightarrow c_2) = funct(c_1) \cup funct(c_2)$;

$depend(c_1 \leftrightarrow c_2) = depend(c_1) \cup depend(c_2)$;

$descri(c_1 \leftrightarrow c_2) = descri(c_1) \cup descri(c_2)$ 。

性质 1 设构件 $c_i \in U, op_i \in OP, i, j \in Z^+, U, op$ 分别为构件集合和操作集合, $op_1, op_2, \dots, op_n$ 是对应 $c_1, c_2, \dots, c_n$ 的操作动作,则 $op_1 \rightarrow op_2 \rightarrow \dots \rightarrow op_n$ 运算结果是 U 的一个构件。

证明: 假设 $S = (op_1 \rightarrow op_2 \rightarrow \dots \rightarrow op_n)$, 则有: $S_{n-1} = op_1 \rightarrow op_2 \rightarrow \dots \rightarrow op_{n-1}, S_{n-2} = op_1 \rightarrow op_2 \rightarrow \dots \rightarrow op_{n-2}, \dots, S_2 = op_1 \rightarrow op_2$, 根据定义 7, 得到 $S_2, S_3, \dots, S_{n-2}, S_{n-1}$ 都是构件, 而 $S = S_{n-1} \rightarrow S_n$, 所

以是 S 是 U 的一个构件。

类似的还可得到性质 2, 其证明与性质 1 的类似。

性质 2 设构件 $c_i \in U, op_i \in OP, i, j \in Z^+, U, OP$ 分别为构件集合和操作集合, $op_1, op_2, \dots, op_n$ 是对应 $c_1, c_2, \dots, c_n$ 的操作动作, 则 $op_1 + op_2 + \dots + op_n$ 运算结果是 U 的一个构件, $op_1 \leftrightarrow op_2 \leftrightarrow \dots \leftrightarrow op_n$ 运算结果是 U 的一个构件。

定义 11 扩展的基于进程代数的构件运算

$X = T \cup \{\tau\}$ 为全部动作行为的构件集合, τ 表示内部动作, $\alpha \in X, L \subseteq T, r \in R^+$, 定义为 $OP::=0 | (\alpha, r).op | op_1 \rightarrow op_2 | op_1 + op_2 | op_1 ||_L op_2 | op_1 / op_2 | C$

① 0 表示动作终止, 即不执行任何动作;

② $(\alpha, r).op$ 是动作前缀操作, 先执行活动 α 然后执行 op , α 是执行时间为 r 的负指数活动;

③ $op_1 \rightarrow op_2$ 与 $op_1 + op_2$, 意义同定义 7;

④ $op_1 ||_L op_2$ 是并发执行, 表示 op_1 和 op_2 并发执行操作, L 是 2 个都需要执行的动作, 当 L 为空时, 记为 $op_1 || op_2$;

⑤ op_1 / op_2 表示 op_1 对 op_2 活动隐藏, op_1 / op_2 中那些在集合 op_1 中的动作被看成内部动作, 不能被外部观察到;

⑥ C 代表常量, 定义方程 $C = OP$, C 具有与 OP 同样行为, 允许递归定义。

如果没有括号来确定操作优先级, 默认的从高到底优先级排列: $., \rightarrow, +, ||, \setminus$ 。

2.3 网构软件代数系统

由于任何一个程序都可用顺序、选择、循环三种结构组合实现^[7]。

定义 12 设论域 $U, S = (I, A, R)$ 为网构软件系统, 其中 I 表示网构软件构件集合, A 表示构件运算, $A \in \{\rightarrow, +, \leftrightarrow\}$ 。

定理 1 由选择、循环和顺序运算可以构成一个代数系统。

证明: 根据性质 1、2, 构件选择、循环和顺序运算结果还是同一个构件论域下的一个构件, 所以在以上三种运算下是一个代数系统。

定义 13 构件运算系统的完整性

① $c_i \in U, \forall c_i$ 具有某一特定功能, 且 $(c_i A c_j) \in U, A \in \{\rightarrow, +, \leftrightarrow\}$ 。

② 论域 U 的 $\forall c_i$ 上的 $A \in \{\rightarrow, +, \leftrightarrow\}$, 都是封闭的。

③ 构件运算关系生成的网构软件系统是封闭

的。

定理 2 构件运算系统是一个完整的代数系统, 可以描述任意网构软件系统的体系结构。

证明: 根据定义 1, 每个构件包括构件与其它构件具有互操作关系, 显然, $\forall c_i$ 具有一功能, 且 $(c_i A c_j) \in U, A \in \{\rightarrow, +, \leftrightarrow\}$, 所以构件 C 是完整的。

由性质 1、性质 2 可知, 构件的三种运算结果还是原论域的一个构件, 所以同论域上构件对每一种运算是封闭的。

任何一个程序都可用顺序、选择、循环三种结构实现, 因此任何一个基于构件运算的软件体系结构可以分解为顺序、选择和循环组合运算而成。根据性质 1、2, 通过构件运算关系生成的软件系统是封闭的, 因此构件运算系统是一个完整的代数系统。

综上, 构件运算关系可以描述任意软件系统的体系结构, 构件运算系统是一个完整的代数系统, 并可以描述任意网构软件系统的体系结构。

定理 3 构件运算描述的网构软件体系结构具有封闭性、层次性和可扩充性。

证明: 定理 2 的证明可知, 构件运算系统是封闭的, 故其描述的体系结构也是封闭的。

网构软件体系结构可由构件通过构件运算关系连接而成, 根据性质 1、2, 组合运算形成的还是构件, 那么通过构件运算重新组合后可得到更大的软件体系结构, 因此其具有层次性, 当然也有扩充性。

构件动作并发交替进行, 是一个交织语义模型:

$$op_1 ||_L op_2 = \sum_{a_i \notin L} a_i. (op_1 ||_L op_2) + \sum_{b_j \notin L} b_j. (op_1 ||_L op_2) + \sum_{a_i = b_j \in L} a_i. (op_1 ||_L op_2) \quad (1)$$

在交织语义模型下, 任何进程代数表达式都可以展开为仅包含前缀和选择操作的形式, 即

$$\sum_{a^i \in ACT} a_1^i \cdot a_2^i \cdot \dots \cdot a_n^i \cdot 0 = a_1 \cdot a_2 \cdot \dots \cdot a_n \cdot 0 + b_1 \cdot b_2 \cdot \dots \cdot b_n \cdot 0$$

引理 网构软件系统可以用顺序和选择算子组成的进程代数表达式表示。

证明: 由式(1)知, 构件并发可用仅包含前缀和选择操作的形式表达, 顺序和选择是进程代数基本算子, 所以网构软件系统是可以顺序和选择算子组成的进程代数表达式表示。

3 网构软件可靠性评估

3.1 网构软件系统可靠性计算

定义 14 构件可靠度组合运算, 设网构软件系

统 S 由构件 $t_1, t_2, \dots, t_n$ 组成, 构件可靠度分别 $R(c_1), R(c_2), \dots, R(c_n)$, 为了简单, 假设每个构件只执行一次, 则构件可靠度运算如下:

① 构件顺序操作可靠度:

$$R(S) = \prod_{i=1}^m R(c_i) \quad (2)$$

② 构件选择操作, 构件执行概率 $p(c_i)$ 时可靠度为:

$$R(S) = \prod_{i=1}^m [R(c_i) p(c_i)] \quad (3)$$

③ 构件循环顺序操作, 如果循环体 $A \in C, C$ 为构件集合, A 执行 t 次, 系统可靠度 $R(S) = \prod_{i=1}^t R(A)$; 如 A 执行到特定条件停止, 流程控制返回 A 的概率 $P(A)$, 停止并跳出循环的概率为 $1 - P(A)$, 那么可靠度为:

$$R(S) = (1 - p(A))R(A) / (1 - R(A)p(A)) \quad (4)$$

定义 15 网构软件系统可靠度是所有构件成功运行的概率, 即运行成功的期望度, 设网构软件系统 S 由构件 $c_1, c_2, \dots, c_n$ 构成, 每个构件在系统中成功运行次数变量分别为 $t_1, t_2, \dots, t_n$, 每个构件的可靠度分别是 $R(c_1), R(c_2), \dots, R(c_n)$ 组建, 假设构件的失效均相互独立, 则系统的可靠性表示为:

$$R(S) = E(\prod_{i=1}^n R(c_i)^{t_i}) \quad (5)$$

函数 $H(x)$, X 为随机变量, $E(X), Var(X)$ 分别为期望和方差, 由泰勒级数展开式:

$$H(x) = H(E(X)) + H'(E(X))(x - E(X)) + (1/2)H''(E(X))(x - E(X))^2$$

近似地: $E(H(X)) = H(E(X)) + H'(E(X))Var(X)$, 那么有 $E[H(X)] \approx H(E(X))$, 因此可靠度简化为计算:

$$\begin{aligned} R(S) &= E(\prod_{i=1}^n R(c_i)^{t_i}) = \prod_{i=1}^n E(R(c_i)^{t_i}) \\ &\approx \prod_{i=1}^n R(c_i)^{E(t_i)} \end{aligned} \quad (6)$$

网构软件系统 S 由构件 $c_1, c_2, \dots, c_n$ 组成, 每个构件功能行为视为一种执行状态, 构件之间转移情况可以得到转移概率矩阵 Q , 因此 Q^k 就是执行 k 步的转移概率, 且 $\lim_{k \rightarrow \infty} \sum_{i=0}^k Q^i = \sum_{i=0}^{\infty} Q^i = (I - Q)^{-1}$, 有 $T = (I - Q)^{-1} = I + Q + Q^2 + \dots = \sum_{i=0}^{\infty} Q^i$. $t_{i,j} = E[X_{i,j}]$. $X_{i,j}$ 表示从 i 到 j 的次数, $t_{i,j}$ 表示 $X_{i,j}$ 的期望, 所以对于构件 c_i , $E(t_i) \approx T_{1i}$, 因此系统可靠度还可以简化为近似计算:

$$R(S) = \prod_{i=1}^n [R(c_i)]^{T_{1i}} \quad (7)$$

3.2 网构软件可靠性评估步骤

(1) 根据网构软件模型形式化网构软件体系结构;

(2) 将系统从顶到下简化成顺序、选择和循环基本结构, 部分构件可能简化需要组合成中间临时构件, 并分别计算临时构件;

(3) 根据网构软件系统结构和构件可靠度或者临时构件可靠度, 计算网构软件系统可靠度, 还可以结合各个构件敏感度, 进行网构软件系统的可靠性整体评估.

4 实验验证分析

(1) 设构件 $C1, C2$ 组成简单的网构软件, 可靠性分别为 $R1, R2$, 在构件系统中, 构件 1 要么调用构件 1 一次进行分支选择, 或以概率为 p 调用构件 2, 或者已概率 $1 - p$ 不调用回到 $R1$, 而构件 2 不调用任何构件. 可以得到构件运行序列: $C1, C1C2C1, C1C1C2C1C2, \dots$, 形式化系统可靠性代数表示:

$$\begin{aligned} &R1 \times (1 - p) + R1 \times (1 - p) \times R2 \times p \times R1 + \\ &R1 \times (1 - p) \times R2 \times p \times R1 \times R2 \times p \times R1 + \dots \\ &R1 \times (1 - p) \times R2 \times p \times R1 + R1 \times (1 - p) \times R2 \times p \times R1 \times \\ &R2 \times p \times R1 \times R2 \times p \times R1 + \dots = \sum_{i=1}^{\infty} (1 - p) R1 \\ &(R2pR1)^{i-1}. \end{aligned}$$

得到系统可靠度:

$$\begin{aligned} R &= \sum_{i=1}^{\infty} R1^i R2^{i-1} p^{i-1} (1 - p) \\ &= (1 - p) \sum_{i=1}^{\infty} R1^i R2^{i-1} p^{i-1} \\ &= (1 - p) R1 \sum_{i=1}^{\infty} R1^{i-1} R2^{i-1} p^{i-1} \\ &= (1 - p) R1 \frac{1}{1 - pR1R2} \end{aligned}$$

而按照传统的 Cheung 模型方法, 得到基础矩

$$\text{阵 } T = \begin{bmatrix} 1 & pR1 & (1-p)R1 \\ R2 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, k=3, \text{可靠度为:}$$

$$R = (-1)^{k+1} (|(I - T)_{k,1}|) / (|I - T|) = (1 - p)R1 / (1 - pR1R2)$$

可见, 二者计算结果一样.

(2) 某系统如图 1 所示. 构件可靠性和转移率分别为: $R_1 = 0.998, R_2 = 0.99, R_3 = 0.98, R_4 = 0.995, R_5 = 0.999, R_6 = 0.985, R_7 = 0.996, R_8 = 0.975, R_9 = 0.99, R_{10} = 0.998, R_{11} = 0.95, R_{12} = 0.965, R_{13} = 0.97, R_{14} = 0.98, R_{15} = 0.992; p_{1,2} =$

$p_{13,14}=0.4$, $p_{1,3}=p_{1,4}=p_{13,15}=0.6$, $p_{2,6}=p_{3,7}=p_{4,7}=p_{5,8}=p_{6,7}=p_{14,15}=1$, $p_{6,9}=p_{6,10}=0.9$, $p_{7,11}=0.25$, $p_{8,11}=0.8$, $p_{7,9}=p_{7,10}=0.75$, $p_{8,5}=p_{11,8}=0.2$, $p_{9,12}=p_{10,12}=0.7$, $p_{11,13}=p_{12,2}=p_{12,13}=0.5$, $p_{9,13}=p_{10,13}=0.3$, $p_{11,1}=p_{11,14}=0.15$.

计算得到 $T(1,i)=(1.030 \ 0.8291 \ 0.6168 \ -0.1055 \ 0.8208 \ 0.6823 \ -0.5275 \ 1.237 \ 0.2110 \ 0.8661 \ 0.8893 \ 0.3751 \ 0.8852)$, 其中 $i=1,2,\dots,13$,

采用公式(7)得到 $R(S)=\prod_{i=1}^n [R(C_i)]^{T_{1i}}=0.87753378$, 而用传统的 Cheung 模型方法计算结果为 0.8781, 二者相差很小. 可见, 提出的可靠性计算方法是正确和有效的.

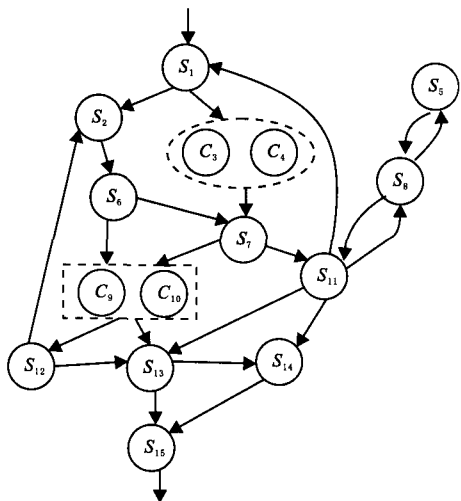


图1 系统状态图

5 结束语

本文从定性和定量两方面分析和研究网构软件,进行了基于代数理论的形式化分析,建立了网构软件的模型和演算框架,研究了网构软件体系结构和构件组合运算,讨论了基于网构软件模型上的代数系统和推演,并提供了网构软件可靠性的简化计

算方法. 实验证明建立的模型和提出的方法正确、有效. 下一步将在本文研究基础上,开展网构软件服务发现、应用组合以及演化性方面的可靠性研究.

参考文献:

- [1] Zhao Hui Qun Sun Jing. An algebraic model of service oriented trustworthy software architecture[J]. Chinese Journal of Computers, 2010,33(5):890 - 899.
- [2] 赵会群,孙晶. 面向服务的可信软件体系结构代数模型[J]. 计算机学报,2010,33(5):890 - 899.
- [3] 胡海洋,吕建. 面向网构软件的构件组装可靠性分析[J]. 中国科技论文在线,2006,1(3):165 - 174.
- [4] Wang Ping, Sun Changsong, Li Lijie. Primary research on Internetwork reliability technology[C]// International Multi - Symposiums on Computer and Computational Sciences. China, Hangzhou, 2006:424 - 428.
- [5] Du Yugen, Li Yonggang. Research on internetwork reliability measurement model based on service update [J]. Journal of Frontiers of Computer Science and Technology, 2008, 2(4):431 - 438.
- [6] Yan Guiling. Model for tectonic method of reliable Internetwork based monitoring [C]// International Conference on Computer Science and Network Technology. China, Beijing, 2011:2429 - 2432.
- [7] Bohm G, Jacopini G. Flow diagrams, turning machines and languages with only two formation rules[J]. Communication of the ACM, 1966, 9(5):366 - 371.

作者简介:

张 靖 男, (1972—), 博士研究生. 研究方向计算机软件可靠性和计算机网络工程.

雷 航 男, (1960—), 教授, 博士生导师. 研究方向为实时软件工程和软件可靠性工程.