

构件化业务流程重组应用服务器可信性度量方法研究

韩强^{1,2,3}, 袁玉宇^{1,2}

(1. 北京邮电大学 计算机学院, 北京 100876;

2. 北京邮电大学 可信分布式计算与服务教育部重点实验室, 北京 100876;

3. 北方民族大学 计算机科学与工程学院, 宁夏 银川 750021)

摘要: 针对面向业务流程重组的应用服务器 (BPRAS, business process re-engineering oriented application server) 的可信性度量方法形式化方面的不足, 给出了 BPRAS 的业务流程建模与可信性度量协作框架, 并在此框架基础上, 提取组成应用服务器的构件属性和构件组合行为特征, 通过进程代数等相关方法进行抽象, 从构件之间的运算角度, 定义 BPRAS 构件运算算子, 从而建立 BPRAS 代数模型, 并进一步对其支撑的业务流程应用软件可信性进行建模, 提出多种可信范式, 为可信 BPRAS 软件可信性度量方法提供理论支撑。最后介绍了一个应用实例。

关键词: 业务流程重组; 应用服务器; 构件; 软件可信性; 度量

中图分类号: TP311

文献标识码: A

文章编号: 1000-436X(2014)03-0047-11

Research on trustworthiness measurement approaches of component based BPRAS

HAN Qiang^{1,2,3}, YUAN Yu-yu^{1,2}

(1. School of Computer Science, Beijing University of Posts and Telecommunications, Beijing 100876, China;

2. Key Laboratory of Trustworthy Distributed Computing and Service, Ministry of Education, Beijing University of Posts and Telecommunications, Beijing 100876, China;

3. School of Computer Science and Engineering, Beifang University of Nationalities, Yinchuan 750021, China)

Abstract: Aiming at the absence of measurement methods about formalization on trustworthiness attributes of business process reengineering oriented application server (BPRAS), a cooperation framework of business process modeling and trustworthiness measurement based on BPRAS was proposed. Extracting the features of characteristics and behaviors of components consisting of the application server, an abstraction view of computing between those components based on the framework was presented through process algebraic approaches. Furthermore, relative concepts about its components computing operators were defined in order to construct algebraic model of its software trustworthiness (ST) to support the measurement of trustworthy BPRAS. Finally, an application case was also given.

Key words: business process reengineering; application server; component; software trustworthiness; measurement

1 引言

由于业务流程及 workflow 系统的相关理论、方法与技术, 在农业、工业和服务业信息化领域广泛的应用, 在 workflow 系统支撑下, 业务流程应用软件所具

备的较高灵活性和可维护性等优点已被用户广泛接受, 在此基础上, 用户也对其可信性提出了更高的要求。根据业务需求变化, 原有业务流程不可避免地需要重组、更新和优化, 因此业务流程重组 (BPR, business process re-engineering) 在近年成为

收稿日期: 2012-10-25; 修回日期: 2013-11-15

基金项目: 国家自然科学基金资助项目(61363001, 91118002, 71061001, 61162013); 国家民委科研基金资助项目(12BFZ010); 北方民族大学科研基金资助项目(2013XYZ030)

Foundation Items: The National Natural Science Foundation of China (61363001, 91118002, 71061001, 61162013); The Science Foundation of State Nationalities Affairs Commission (12BFZ010); The Science Foundation of Beifang University of Nationalities (2013XYZ030)

该领域的研究重点，而如何采集、度量和评估面向 BPR 的应用服务器及应用软件可信性，是该研究领域一项值得研究的重要课题。

通常意义下，可信软件是指在特定环境下其运行行为及其结果符合人们预期，并在受到干扰时仍能提供连续服务的软件^[1]。当前学术界和产业界从不同角度，采用不同方法对可信软件相关理论展开大量研究，同时作为研究成果，已有大量相应工具出现。但是上述成果大部分关注于通用软件可信性研究^[2-4]，且还未形成较统一的理解和认识，因而对包括业务流程重组应用服务器（BPRAS, BPR oriented application server）在内的特定领域软件可信性指导性和适用性尚显不足。

基于以上研究背景，本文重点针对 BPRAS 这一特定领域软件，以业务流程可信性度量为该类领域软件问题域，以该类领域软件所支撑的高可信业务流程应用软件为解域，深入研究业务流程应用软件可信性度量的形式化方法，对可信业务流程应用软件进行建模，从而提高业务流程应用软件可信性，为可信 BPRAS 软件设计提供理论模型支撑，并为相关工具软件研发提供方法与技术支持。

具体而言，本文采用以下方法完成上述研究工

作。首先给出 BPRAS 元构件及其派生的构件形式化定义，并在此基础上构建构件运算符，进而基于构件定义和构件算子给出构件组合的形式化定义；其次，基于构件组合和进程代数理论，建立 BPRAS 代数模型，并给出基于 BPRAS 的业务流程应用软件可信性模型的形式化定义，以及由多种可信范式构成的可信业务流程应用软件规范化等级理论模型。

毫无疑问，构造可信业务流程应用软件需要系统性的理论与方法，仅依靠 BPRAS 自身难以完成可信业务流程重组应用软件的研发任务，其关键原因在于 BPRAS 处理的是已完成重组的业务流程，而重组后的业务流程是否满足用户业务流程重组需求需要流程开发工具出具验证。另一方面，经过对业务流程可信性分析后确定的规范化等级可反馈至流程开发工具，供用户参考，并完善业务流程至更高规范化等级。由此形成业务流程重组应用服务器与基于语义验证的可视化流程建模工具^[5]和业务流程可信性度量协作框架(如图 1 所示)。根据该框架，对于 BPRAS 的业务流程建模与可信性度量协作机制的描述如下。

1) 需求分析人员编制新业务流程（如图 1①所

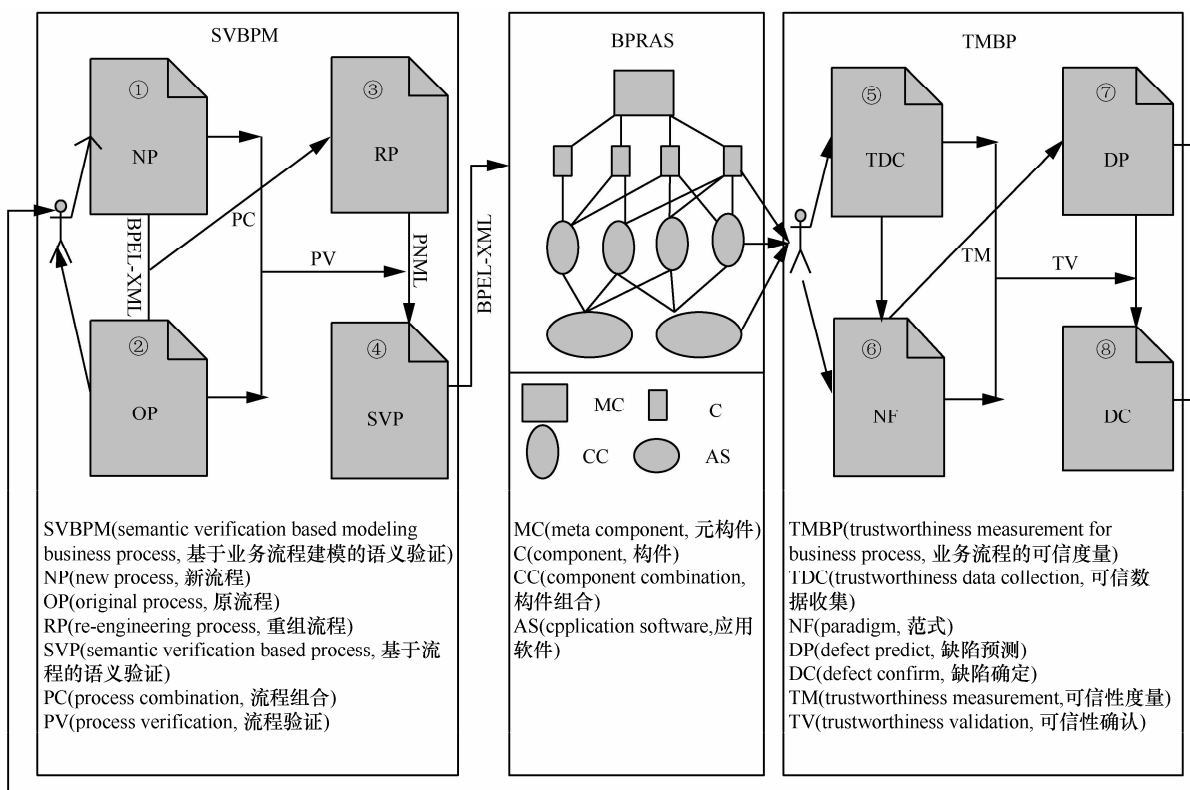


图 1 BPRAS 的业务流程建模与可信性度量协作框架

示), 并修改调整原业务流程(如图1②所示)。

2) 需求分析人员根据业务流程重组需求编制重组业务流程(如图1③所示)。

3) 基于语义的业务流程验证工具自动将重组业务流程从XML描述的BPEL业务流程转换为Petri Net描述的业务流程, 并在验证后形成出具验证业务流程(如图1④所示)。

4) 根据出具验证业务流程, 通过适当的前端开发工具完成其在业务流程重组应用服务器, 即工作流系统中部署。部署后形成的业务流程应用软件由元构件派生的构件及构件组合共同组成。并在实际运行中, 由可信性采集系统(如图1⑤所示)实时采集构件个体及应用软件的可信性数据。

5) 根据应用软件体系结构(如图1⑥所示)及构件个体采集数据, 通过可信性度量查找影响应用软件可信性的主要构件或构件组合, 对这类构件或构件组合提出缺陷预测(如图1⑦所示), 并将缺陷预测记录入库, 支持缺陷判定(如图1⑧所示)。

6) 通过将缺陷预测反馈给需求分析人员, 由其修改调整业务流程, 形成新的业务流程, 在重新采集新业务流程可信性后, 由缺陷判定缺陷预测的主要构件或构件组合对应用软件可信性影响的结论。

2 相关工作介绍

业务流程重组是服务计算领域一个非常重要的研究分支, 国内外已有许多学者针对与业务流程执行紧密相关的工作流系统的理论、方法和技术展开研究。在BPR相关的工作流理论方法及技术方面, 文献[6,7]介绍了工作流核心技术及发展趋势以及利用上述技术构成的工作流管理系统及其概念、分类和参考模型。在BPR相关理论方法研究工作中, 曹军威等人提出一种基于 π 演算状态算子的域解析算法用于网格工作流数据抽象、管理、建模、时序验证及复杂性降解^[8]。在考虑队列等待时间和平衡数据移动成本所造成性能影响的前提下, 余志勇等人提出一种多簇计算环境下动态工作流调度算法^[9]。文献[10]基于时态工作流元模型提出一种综合描述过程、资源、案例和时间四维信息的时态工作流过程模型TPWF-net, 并给出了相应的合理性验证方法。上述文献重点从工作流系统的研发角度针对其通用质量展开研究, 较少涉及工作流系统的用户在使用过程中对其系统行为可信性的度量,

且不涉及工作流系统支撑的业务流程重组过程与其建立的业务流程可信性之间的关联关系。

在可信工作流理论方法及技术方面, 文献[11]针对可信工作流的标准研究方面的不足, 从信任在工作流系统中承担的角色及其所处的上下文环境对其承担角色的影响进行了分析和讨论。通过综合评价可靠性、可用性及安全性等属性, 文献[12]提出了一种粒子群优化算法, 该算法可用于网格环境下工作流调度决策支持。针对仅从QoS角度衡量工作流的服务质量的片面性问题, 文献[13]将参与服务请求和服务提供双方的信任关系作为QoS调度的重要指标之一, 提出一种基于信任关系Web服务工作流的QoS调度模型。工作流的可信性在一定程度上依赖于其所使用的资源是否能够满足工作流服务的需求, 为提高资源符合工作流服务的特定需求, 文献[14]对传统的基于QoS的服务提供方选择资源的方式进行了改进, 加入了提供方对资源的信任评估过程, 使得更加适宜工作流特定需求的资源被选入工作流的执行过程, 从而提高了工作流的执行效率, 也避免了资源的浪费。

此外, 国内一些学者针对工作流相关的核心技术进行了研究, 例如文献[15]基于业务流的网络流量提出一种分段采样自适应方法, 提高对其的监测管理能力, 该方法可用于业务计费和流量工程, 也可为业务流程的通信可信性提供可参考的技术手段。文献[16]以工作流模型的语义验证技术为研究重点, 通过对其表达业务规则的语义约束集形式化, 将工作流验证问题转化为其语义完整性验证问题, 但该方法是针对工作流模型的静态验证方法, 不属于工作流的运行时语义验证, 即不包含其实时行为的验证工作, 同时也未涉及构成工作流的业务流程重组的语义验证问题。监测提出后续相关工作包括BPRAS的可信性研究与应用等。从工作流使用的资源层风险对整个业务流程的安全属性影响的角度, 文献[17]提出一种层次化的面向业务的风险评估模型, 该模型可以作为本文所提方法的参考验证模型, 但需要进一步研究两类方法的集成技术。任何工作流系统的执行过程可信性均包含了时序约束, 即在预定的时间范围成功完成既定任务, 从对工作流的时序约束角度, 文献[18]提出了一种基于时序逻辑和模型检查的工作流时序建模与一致性验证方法, 该方法可有效地支持工作流系统支持的业务流程时序建模与验证, 但该方法仍然是面

向业务流程的静态验证方法,不涉及在实际运行环境中,受环境中各种因素的影响,其业务流程所体现的行为时序可信性问题。

通过对以上相关文献的分析和总结,可以看到,当前多数有关 workflow 可信性评价的工作集中在 workflow 提供的服务端可信度评价方面展开研究,而针对 workflow 的支撑运行平台,尤其是业务流程重组应用服务器的运行时行为可信性研究较少,这造成了以上研究工作多以可信性评估为主,而很少涉及可信性度量的问题。本文以业务流程的运行行为与其预期声明的一致性为度量模型的核心准则,以该类服务器平台的软件体系结构的范式模型为度量基础,通过进程代数等相关方法给出了其可信性度量方法,并结合具体的业务流程重组应用实例研究分析了业务流程可信性的特点和规律。

3 BPRAS 构件模型

3.1 元构件与分层构件

3.1.1 元构件

定义 1 元构件可定义为四元组 $C = \langle id, type, body, envi \rangle$ 。

id 是构件的标识。

$type = \{I; II; III; \dots\}$ 是构件类型。

$body = (I, O, E : \{E^I; E^{II}; E^{III}; \dots\})$ 是构件运行体; 执行体 E 的各种枚举类型分别对应各种构件类型。

$envi = (c_type, c_body)$ 是构件通信环境体; 其中, c_type 是通信协议类型, c_body 是通信构件执行体。

3.1.2 用户层构件

定义 2 用户层构件可定义为四元组 $C^U = \langle U_id, U_type, U_body, U_envi \rangle$ 。

U_id 是构件的标识。

$U_type = \{I; II; III\}$ 是构件类型; 其中, I 型代表与 BPRAS 格式保持一致, 系统需设置报文格式解析模块; II 型代表受 BPRAS 格式控制, 系统需设置报文格式解析模块; III 型代表独立于 BPRAS 设计报文格式系统。

$U_body = (I, O, E : \{Split; Explain; Translate\})$ 是构件运行体; 执行体 E 的 3 种枚举类型分别对应 $I \sim III$ 3 种构件类型。

$U_envi = (c_type, c_body)$ 是构件通信环境体。

3.1.3 服务层构件

定义 3 服务层构件可定义为四元组 $C^S = \langle S_id, S_type, S_body, S_envi \rangle$ 。

S_id 是构件的标识。

$S_type = \{I; II; III\}$ 是构件类型; 其中, I 型代表通信子层; II 型代表平台子层; III 型代表应用子层; $S_body = (I, O, E : \{Comm; Plat; Appl\})$ 是构件运行体; 执行体 E 的 3 种枚举类型分别对应 $I \sim III$ 3 种构件类型。其中,

$Comm = (I, O, E : \langle AsyLsn, AsyAns \rangle; SynCom)$

即执行体 E 的通信子层 $Comm$ 同构于其母体执行体 E , 且自身执行体 E 的 2 种枚举类型分别为异步通信 $\langle AsyLsn, AsyAns \rangle$ 和同步通信 $SynCom$ 模块。

$Plat = (I, O, E : (MainControl; \langle Secu; Tran; Fund \rangle))$

即执行体 E 的平台子层 $Plat$ 同构于其母体执行体 E , 且自身执行体 E 的 2 类模块分别为主控模块 $MainControl$ 和辅助模块 $\langle Secu; Tran; Fund \rangle$, 即安全控制; 格式转换和基本功能支撑 3 种模块。

$Appl = (I, O, E : (Business; Mech; Common))$

即执行体 E 的平台子层 $Plat$ 同构于其母体执行体 E , 但是应用子层特殊之处在于, 其输入来自于 BPR 配置工具出具验证后的业务流程, 其输出为业务流程执行日志, 且自身执行体 E 的 3 类配置模块分别为业务流程配置模块 $Business$, 商户配置模块 $Mech$ 和公共配置模块 $Common$ 。实际上, 上述 3 类配置模块并无执行代码, 仅为配置信息。

$S_envi = (c_type, c_body)$ 是构件通信环境体。

3.1.4 资源层构件

定义 4 资源层构件可定义为四元组 $C^R = \langle R_id, R_type, R_body, R_envi \rangle$ 。

R_id 是构件的标识。

$R_type = \{I; II; III\}$ 是构件类型。其中, I 型代表 INSERT 解析模块; II 型代表与 UPDATE 解析模块; III 型代表 DELETE 解析模块。

$R_body = (I, O, E : \{Insert; Update; Delete\})$ 是构件运行体; 执行体 E 的 3 种枚举类型分别对应 $I \sim III$ 3 种构件类型。

$R_envi = (c_type, c_body)$ 是构件通信环境体。

由上述定义可知 BPRAS 各层均同构于元构件 C 。即 $\{C^U \cong C^S \cong C^R \cong C\}$ 。

3.2 BPRAS 构件算子

在本节中, 设上述分层构件全体组成论域为 U 。

3.2.1 使用算子

定义5 设 A, B 是论域 $Dom(U)$ 中的2个构件, 若

$$(\exists x \in (A) \cdot O, \exists y \in (B) \cdot I, \exists a \in (A) \cdot I, \exists b \in (B) \cdot O) \wedge ((y \subseteq x) \rightarrow (B) \cdot E) \wedge ((B) \cdot E \rightarrow (a \subseteq b))$$

即构件 A 通过“使用”构件 B 的 I 实现其内部 E 代表的功能需求, 则称 A, B 进行了一次“使用”运算, 记作 $A \downarrow B$ 。“使用”是最基本的集中式构件组合运算。

3.2.2 激发与反馈算子

定义6 设 A, B 是论域 $Dom(U)$ 中的2个构件, 若

$$(\exists x \in (A) \cdot O \wedge \exists y \in (B) \cdot I \wedge (y \subseteq x)) \wedge ((A) \cdot env_i \cdot c_type = (B) \cdot env_i \cdot c_type) \wedge ((A) \cdot env_i \cdot c_body \rightarrow (B) \cdot env_i \cdot c_body)$$

即构件 A 通过同类协议向构件 B 发送一个消息“激发”构件 B 中的 E , 并通过 I 接收 A 发出的服务请求, 则称 A, B 进行了一次“激发”运算, 记作 $A \mapsto B$ 。

定义7 设 A, B 是论域 $Dom(U)$ 中的2个构件, 若

$$(\exists x \in (A) \cdot I \wedge \exists y \in (B) \cdot O \wedge (y \subseteq x)) \wedge ((A) \cdot env_i \cdot c_type = (B) \cdot env_i \cdot c_type) \wedge ((B) \cdot env_i \cdot c_body \rightarrow (A) \cdot env_i \cdot c_body)$$

即构件 B 通过同类协议向构件 A “反馈”一个消息, 则称 A, B 进行了一次“反馈”运算, 记作 $A \prec B$ 。“激发”与“反馈”是最基本的分布式构件组合运算, 将一组“激发”与“反馈”组成的运算对, 即 $(A \mapsto B) \rightarrow (A \prec B)$, 称为“调用”运算, 简记为 $A \downarrow B$ 。

3.2.3 协同与并行算子

定义8 设 A, B 是论域 $Dom(U)$ 中的2个构件, 若

$$(\exists x \in (A) \cdot I, \exists y \in (B) \cdot O) \rightarrow ((x \rightarrow (A) \cdot E) \rightarrow ((B) \cdot E \rightarrow y)) \wedge (\exists x \in (A) \cdot O, \exists y \in (B) \cdot I) \rightarrow ((y \rightarrow (B) \cdot E) \rightarrow ((A) \cdot E \rightarrow x))$$

即构件 A 的运行必导致构件 B 的运行, 反之亦成立, 则称 A, B 是“协同”运算, 记作 $A \equiv B$ 。

定义9 设 A, B 是论域 $Dom(U)$ 中的2个构件,

若 $(A \equiv B) \wedge ((A) \cdot I \wedge (B) \cdot I = \emptyset)$, 即构件 A 与构件 B 在无依赖关系前提下以完全独立运行方式“协同”运算, 则称 A, B 是“并行”运算, 记作 $A \parallel B$ 。

3.2.4 重复与选择算子

定义10 设 A, B 是论域 $Dom(U)$ 中的2个构件, 若 $\exists x_1 \in (A) \cdot I, \exists y_1 \in (A) \cdot O, \exists x_2 \in (B) \cdot I, \exists y_2 \in (B) \cdot O$, 若 $(y_2 \rightarrow x_1) \Leftrightarrow (y_1 \rightarrow x_2)$, 即构件 A 与构件 B 在满足运行条件下, 相互触发对方, 则称 A, B 进行了一次“重复”运算, 记作 $A \odot B$ 。特别地, 当 $A = B$ 时, 称 A 重复执行, 简记为 $\odot A$ 。

定义11 设 A, B 是论域 $Dom(U)$ 中的2个构件, 若 $\exists x_1 \in (A) \cdot I, \exists y_1 \in (A) \cdot O, \exists x_2 \in (B) \cdot I, \exists y_2 \in (B) \cdot O$, 使得 $((x_1 \rightarrow y_1) \wedge \neg x_2) \vee (\neg x_1 \wedge (x_2 \rightarrow y_2))$, 即构件 A 和构件 B 在满足运行条件下, 有且仅有一个构件可获得执行权, 则称 A, B 是“选择”运算, 记作 $A \oplus B$ 。

3.3 BPRAS 构件组合

定义12 BPRAS 构件组合是上述 BPRAS 构件运算的实现。在 BPRAS 中构件组合有特定的语义, 即它是在平台子层 $Plat$ 执行体 $Exec$ 的主控模块 $MainControl$ 控制下, 根据出具验证的 BPR 流程完成的一次构件运算。可定义为五元组 $\langle Id, Step, Beha, Logs, Envi \rangle$ 。

Id 是组合标识。

$Step$ 是组合与构件交互点的序列集合, 每个 $Step = \langle Sid, Sname, Scode, Stype, Senvi \rangle$ 。其中, Sid 是 $Step$ 标识, $Sname$ 是 $Step$ 名称, $Scode$ 是 $Step$ 代码, $Stype$ 是 $Step$ 类型, $Senvi$ 是 $Step$ 环境变量的集合。

$Beha$ 是组合行为语义描述。

$Logs$ 是组合中各个 $Step$ 的行为日志描述集合。

$Envi$ 是组合环境变量的集合;

4 BPRAS 应用软件可信度量

4.1 BPRAS 应用软件体系结构

BPRAS 是支撑 BPR 应用软件运行的平台软件, 其体系结构已在上节给出其构件及构件组合的形式化描述, 下面给出其所支撑的 BPR 应用软件体系结构代数模型。

定义13 设全体 BPR 应用软件组成论域为 U , 则:

- 1) BPR 构件是一个 BPR 应用软件体系结构;

2) BPRAS 构件组合是一个 BPR 应用软件体系结构;

3) BPRAS 构件经由有限次组合 (构件运算) 后是一个 BPR 应用软件体系结构。

BPR 应用软件体系结构记为 $AS = \langle C, O \rangle$, 简称应用软件。其中, $C \subseteq \{C^P, C^F, C^D\}$ 表示组成应用软件的构件集合, $O \subseteq \{\dashv, \vdash, \prec, \succ, \downarrow, \Rightarrow, \parallel, \odot, \oplus\}$ 表示构件运算的集合。可以证明 AS 对任意一个运算构成代数系统, 具备封闭性, 即

$$\forall c_1 \in C, \forall c_2 \in C, \forall o \in O, c_1 o c_2 \in C$$

将 $AS = \langle C, O \rangle$ 称为 BPR 应用软件的代数模型, 也称为 BPR 应用软件的代数表达式。

4.2 BPRAS 应用软件可信性模型

如引言所述, 当前国际上尚不存在统一认可的软件可信性概念标准, 但是“软件的行为符合其预先的声明”, 是度量软件行为是否可信的共识。基于该共识, 给出以下概念。

定义 14 设 $AS = \langle C, O \rangle$ 是一个应用软件, 若 $((\exists c \in C)((c) \cdot I = 1) \Leftrightarrow ((c) \cdot O = 1))$, 则称 c 为 AS 的一个可信构件, 简记为 $T(c) = 1$, 若 $((\forall c \in C)T(c))$, 则称 AS 是一个可信应用软件, 简记为 $T(AS) = 1$ 。

定义 15 设 $AS = \langle C, O \rangle$ 是一个应用软件, 若 $((\exists c \in C)((c) \cdot I = 1) \rightarrow ((c) \cdot O = 0))$, 则称 c 为 AS 的一个失信构件, 简记为 $F(c) = 0$, 若 $((\forall c \in C)F(c))$, 则称 AS 是一个失信应用软件, 简记为 $F(AS) = 0$ 。

定义 16 设 $AS = \langle C, O \rangle$ 是一个应用软件, 任何构件 c 在未被度量可信性前, 没有任何证据确定其是否可信或失信。故确定其初始可信性值为 0.5, 则称 c 为 AS 的一个初始可信构件, 简记为 $I(c) = 0.5$, 若 $((\forall c \in C)I(c))$, 则称 AS 是一个初始可信应用软件, 简记为 $I(AS) = 0.5$ 。

定义 17 设 $AS = \langle C, O \rangle$ 是一个应用软件, 根据对 c 或 AS 可信性度量结果, 记其可信性度量值分别为 $t_c = \psi(b^S, b^T)$ 和 $t_{AS} = \zeta(AS) = \zeta(\forall c \in AS | \psi(c))$, 显然, $t_c, t_{AS} \in [0, 1]$ 。其中, ψ 和 ζ 分别是 c 和 AS 的度量算子, 且是可替换的, 替换规则取决于具体的度量方法; b^S, b^T 分别是 c 实际成功完成的行为向量和 c 预先声明完成的行为向量。

定义 18 设 $AS = \langle C, O \rangle$ 是一个应用软件, 若 $(\exists c_1, c_2 \in C, \forall o \in O)(c_1 o c_2 \in C) \rightarrow (F(c_2) \rightarrow F(c_1))$ 。

即对任意 2 个构件 c_1, c_2 之间所有运算有: 若 c_2 失信, 则 c_1 失信, 则称构件 c_1 可信性依赖于 c_2 , 简记为 $c_2 \Rightarrow c_1$ 。

推论 1 若 $c_2 \Rightarrow c_1$, 且 c_1, c_2 的预先声明完成的行为向量 $b_{c_1}^T, b_{c_2}^T$ 相等, 则 $t_{c_1} \leq t_{c_2}$ 。

证明 首先, 设 ψ 是 c_1 和 c_2 的度量算子, 根据定义 16, $t_c = \psi(b^S, b^T)$, 在 b^T 不变的前提下, $\psi(b^S, b^T)$ 是一个关于 b^S 的单调递增函数。则当 $b_{c_1}^T = b_{c_2}^T$, 由 $c_2 \Rightarrow c_1$ 可得 $b_{c_1}^S \leq b_{c_2}^S$, 因此 $t_{c_1} \leq t_{c_2}$ 。

证毕。

推论 2 在所有构件算子中, 当且仅当 $c_1 \dashv c_2$ 时, 有 $c_2 \Rightarrow c_1$, 且推论 1 成立, 即 $t_{c_1} \leq t_{c_2}$ 。

证明 首先, 证明必要性。当 $c_1 \dashv c_2$ 时, 根据定义 5, 有 $b_{c_1}^T = b_{c_2}^T$, 又因条件 $c_2 \Rightarrow c_1$ 成立, 则根据推论 1, 结论 $t_{c_1} \leq t_{c_2}$ 成立。

其次, 证明充分性。使用反证法。即假设存在算子 $o' \in \{\dashv, \prec, \downarrow, \Rightarrow, \parallel, \odot, \oplus\}$, 使得前提条件 $c_2 \Rightarrow c_1$ 成立。

当 $t_{c_1} \leq t_{c_2}$ 时, 有 $b_{c_1}^T = b_{c_2}^T$, 不失一般性, 不妨设 $o' = \Rightarrow$, 即为激发算子, 则根据定义 6, 有 $(c_1) \cdot O \rightarrow (c_2) \cdot I$, 但该算子不保证 $(c_2) \cdot O \rightarrow (c_1) \cdot I$ 成立, 即若 c_2 失信, 则 c_1 不一定失信, 即 $c_2 \Rightarrow c_1$ 不成立, 显然这与假设矛盾。其他算子类似可以得证。

证毕。

推论 3 使用算子关联的 2 个构件之间的可信性依赖关系具备自反性和传递性。

证明 自反性, 即 $(c_1 \Rightarrow c_1) \rightarrow (c_1 \Rightarrow c_1)$, 显然成立。

传递性, 即 $((c_1 \Rightarrow c_2) \wedge (c_2 \Rightarrow c_3)) \rightarrow (c_1 \Rightarrow c_3)$, 根据定义 17, 若 c_1 失信, 则 c_2 失信, 即 c_3 失信。

证毕。

4.2.1 BPRAS 候选核

定义 19 设 $AS = \langle C, O \rangle$ 是一个应用软件, 若 $(\exists c' \in C) \rightarrow ((\forall c \in C, \exists o \in O) \rightarrow (c o c' \in C))$, 则称 c' 为 AS 的一个候选核, 候选核集合简记为 $Cker(AS)$, 若 $|Cker(AS)| = 1$, 即 AS 只有一个候选核, 则称其为主核, 简称核, 简记为 AS^{ker} 。特别地, 当 $o = \dashv$ 时, 称候选核为可信候选核, 称主核为可信主核, 简称可信核。

定义 20 设 $AS = \langle C, O \rangle$ 是一个应用软件, 若

$(\exists C_1 \subseteq C_{ker}(AS), AS' = \langle C_1, O \rangle)$, 则称 AS' 为 AS 的候选核应用软件。

4.2.2 1NF 范式

定义 21 设 $AS = \langle C, O \rangle$ 是一个应用软件, 若满足 $(\exists c_1 \in C) \rightarrow ((\forall c_2 \in C, \exists o \in O) \wedge (c_2 o c_1 \in C))$, 则称 AS 满足第一范式, 简记为 $1NF$ 。显然, 为满足 $1NF$ 范式, 对软件体系结构设计上的约束等价于物理意义上不允许存在孤立构件, 这是可信软件设计的最低要求。

4.2.3 2NF 范式

定义 22 设 $AS = \langle C, O \rangle$ 是一个应用软件, 若其存在候选核, 则称 AS 满足第二范式, 简记为 $2NF$ 。

4.2.4 3NF 范式

定义 23 设 $AS = \langle C, O \rangle$ 是一个应用软件, 若其存在主核, 则称 AS 满足第三范式, 简记为 $3NF$ 。

4.2.5 BCNF 范式

定义 24 设 $AS = \langle C, O \rangle$ 是一个应用软件, 若 $(\exists c = AS^{ker}, \forall c', c'' \in C, \forall o \in O | c' o c \wedge c'' o c) \rightarrow (c' = c'')$ 即物理意义上, AS 的主核最多拥有 1 个入口运算构件, 则称 AS 满足 $BCNF$ 范式, 简记为 $BCNF$ 。

显然, 上述可信性范式满足以下条件, 即: $BCNF \subseteq 3NF \subseteq 2NF \subseteq 1NF$ 。

4.3 BPRAS 应用软件可信性度量方法

在以上给出 BPRAS 应用软件体系结构、可信性模型基础上, 以下从 BPRAS 组合构件可信性度量算子、基于算子的可信性度量算法和基于范式的可信性度量算法 3 个方面给出 BPRAS 应用软件可信性度量方法。

4.3.1 组合构件可信性度量算子

首先, 从可信性定义 14~16, 并结合 2.2 节中有关算子的语义定义, 以下给出由任意构件算子组合形成的新构件可信性度量值定义。

定义 25

$$M(c_1 o c_2) = \begin{cases} \min(t(c_1), t(c_2)), o = \downarrow \\ t(c_1) \times t(c_2), o = \downarrow \\ \frac{t(c_1) + t(c_2)}{2}, o \in \{\uparrow, \leftarrow, \Rightarrow, \parallel, \odot, \oplus\} \end{cases}$$

即当形成新构件的组合运算为使用算子时, 新构件可信性为参与运算的构件算子可信性最小者, 这是因为根据推论 2, 参与使用算子运算的构件具

备依赖关系, 取二者可信性最小者即可代表新构件可信性; 当组合运算为调用运算时, 新构件可信性为参与运算的构件算子可信性之积, 这是因为根据定义 7, 参与调用算子运算的构件具备影响关系; 其他情况下, 参与运算的构件可信性无相互影响, 为独立运行, 则新构件可信性为参与运算的构件算子可信性的平均值。根据上述分析, 在此确定使用算子为强可信依赖型算子; 调用算子为半可信依赖型算子, 其他算子为弱可信依赖型算子。

4.3.2 基于算子的可信性度量算法

算法 1 基于算子的可信性度量算法

输入: AS

输出: t_{AS}

第 1 步 计算各构件可信性。

第 2 步 根据各算子定义 5~定义 11, 确定运算优先级, 即遍历 $AS = \langle C, O \rangle$, 对其所有运算算子确定优先级从高到底依次为: $\{\downarrow\} > \{\uparrow\} > \{\rightarrow, \leftarrow, \Rightarrow, \parallel, \odot, \oplus\}$ 。

第 3 步 计算优先级最高的使用算子, 根据定义 24, 得到由使用算子构造的组合构件可信性。

第 4 步 计算优先级次高的调用算子, 根据定义 24, 得到由调用算子构造的组合构件可信性。

第 5 步 计算优先级最低的其他构件算子, 根据定义 24, 得到由其构造的组合构件可信性, 最终得到 t_{AS} 。

4.3.3 基于范式的可信性度量算法

由上节给出的算法可以看到, 该算法以基于算子优先级的深度优先搜索计算算法, 其并未考虑应用软件所属范式, 对于同一个组合构件或候选核应用软件同时参与多个其他构件或组合构件运算的情况, 会对其可信性重复计算, 显然增加了计算成本和算法复杂度。基于这方面的考虑, 提出以下基于范式的应用软件可信性度量算法, 其核心思想是首先将应用软件划分为粒度从大到小的 $1NF$ 、 $2NF$ 、 $3NF$ 和 $BCNF$ 范式构成的子应用软件, 每个子应用软件都是一个独立计算区域, 具备构件运算网络复杂度最小的特征, 然后再按照其构成整体应用软件的构造方式, 计算最终的应用软件可信性。

算法 2 1NF 范式可信性计算算法

输入: $AS \in 1NF$

输出: t_{AS}

第 1 步 计算各构件可信性。

第 2 步 根据各算子定义 5~11, 确定运算优先级, 即遍历 $AS = \langle C, O \rangle$, 对其所有运算算子确定优先级从高到底依次为: $\{\downarrow\} > \{\downarrow\} > \{\rightarrow, \leftarrow, \Rightarrow, \parallel, \odot, \oplus\}$ 。

第 3 步 根据各范式定义, 结合上一步算子优先级, 将应用软件划分为由 $2NF$ 组成的独立计算区域。调用算法 2。

第 4 步 根据定义 24, 按照上一步确定的独立计算区域, 依次计算各区域可信性, 最终得到 t_{AS} 。

算法 3 $2NF$ 范式可信性计算算法

输入: $AS \in 2NF$

输出: t_{AS}

第 1 步 计算各构件可信性。

第 2 步 根据定义 20, 将应用软件划分为由 $3NF$ 组成的独立计算区域, 即以每个候选核形成的候选核应用软件 AS' 为独立计算区域, 调用算法 4, 计算其可信性 $t_{AS'}$ 。

第 3 步 根据候选核应用软件组合算子的优先关系, 根据每个候选核应用软件可信性 $t_{AS'}$, 计算其构成的应用软件 AS 的可信性 t_{AS} 。

算法 4 $3NF$ 范式可信性计算算法

输入: $AS \in 3NF$

输出: t_{AS}

第 1 步 计算各构件可信性。

第 2 步 根据主核应用软件组合算子的优先关系, 将应用软件划分为由 $BCNF$ 组成的独立计算区域, 即以每个 $BCNF$ 形成的应用软件 AS' 为独立计算区域, 调用算法 5, 计算其可信性 $t_{AS'}$ 。

第 3 步 根据 $BCNF$ 应用软件之间组合算子的优先关系, 根据每个 $BCNF$ 应用软件可信性 $t_{AS'}$, 计算其构成的应用软件 AS 的可信性 t_{AS} 。

算法 5 $BCNF$ 范式可信性计算算法

输入: $AS \in BCNF$

输出: t_{AS}

第 1 步 分别计算主核和其唯一的入口运算构件可信性。

第 2 步 根据定义 24, 计算其可信性 t_{AS} 。

5 应用研究

本文提出 BPRAS 应用软件代数模型可以直接应用到以下领域: ① BPRAS 自身形式化表示; ② BPRAS 应用软件体系结构设计。在此限于篇幅, 仅介绍在 BPRAS 应用软件体系结构设计中的应用。

为实现上述验证技术, 本文采用的实验环境基于开放平台 Visual Paradigm(一种免费商业 UML 建模工具, 可从 <http://ed.fbk.eu/vlpm> 下载)。基于该平台提供的 API, 开发了出具验证的可视化建模工具, 图 2~图 4 是利用该工具完成后某装备制造铸钢业务流程重组的 3 个验证实例。图中, 第 1 列表示原有业务流程子图, 自上而下构件编号为: $\{Op_s, Op_1, Op_2, Op_3, Op_4, Op_5, Op_6, Op_7, Op_8, Op_9, Op_{10}, Op_e\}$ 。

第 2 列表示删除业务流程子图, 自上而下构件编号为: $\{Op_s, Op_6, Op_7, Op_8, Op_9\}$ 。

第 3 列表示新增业务流程子图, 自上而下构件编号为: $\{Np_1, Np_2, Np_3\}$ 。

第 4 列表示重组业务流程子图, 自上而下构件编号为: $\{Op_s, Op_1, Op_2, Op_3, Op_4, Op_{10}, Op_e, Np_1, Np_2, Np_3\}$ 。

其中, 单箭头线代表使用算子运算; 双向箭头线代表调用运算; 无箭头线代表其他运算。

设图 1~图 3 分别对应 3 种应用软件 $AS1$ 、 $AS2$ 和 $AS3$ 。在经过验证后, 上述 3 种应用软件的重组业务流程分别为

$$AS1 = \langle \{Op_s \downarrow Op_1, Op_1 \downarrow Op_3, Op_1 \Rightarrow Op_2, Op_2 \Rightarrow Op_3, Op_3 \downarrow Op_4, Op_{10} \downarrow Op_3, Op_{10} \downarrow Op_e, Np_1 \downarrow Np_2, Np_1 \parallel Np_3, Np_2 \downarrow Np_3, Op_3 \downarrow Np_2, Np_2 \downarrow Op_{10}\}, \{\downarrow, \downarrow, \Rightarrow\} \rangle$$

$$AS2 = \langle \{Op_s \downarrow Op_1, Op_1 \downarrow Op_3, Op_1 \downarrow Op_2, Op_2 \downarrow Op_3, Op_3 \downarrow Op_4, Op_{10} \downarrow Op_3, Op_{10} \downarrow Op_e, Np_1 \downarrow Np_2, Np_1 \downarrow Np_3, Np_2 \downarrow Np_3, Op_3 \Rightarrow Np_2, Np_2 \Rightarrow Op_{10}\}, \{\downarrow, \downarrow, \Rightarrow\} \rangle$$

$$AS3 = \langle \{Op_s \downarrow Op_1, Op_1 \Rightarrow Op_3, Op_1 \Rightarrow Op_2, Op_2 \Rightarrow Op_3, Op_3 \Rightarrow Op_4, Op_{10} \downarrow Op_3, Op_{10} \downarrow Op_e, Np_1 \Rightarrow Np_2, Np_1 \Rightarrow Np_3, Np_2 \Rightarrow Np_3, Op_3 \Rightarrow Np_2, Np_2 \Rightarrow Op_{10}\}, \{\downarrow, \Rightarrow\} \rangle$$

显然, 在不考虑特殊条件情况下, 任何软件所属构件可信性度量值受其自身开发过程和运行环境等诸多要素综合影响而确定, 不存在唯一的分布规律。在 $AS1 \sim AS3$ 中各有 10 个构件, 在此, 按照随机分布原则, 针对以上 3 种应用软件, 各拟定 100 组该应用软件构件可信性度量值, 并分别执行遵循基于算子可信性度量算法的度量程序, 且按照其均值递增排序为 $t_{AS1}^1 \sim t_{AS1}^{100}, t_{AS2}^1 \sim t_{AS2}^{100}, t_{AS3}^1 \sim t_{AS3}^{100}$ 。下面对图 2~图 4 中的构件见算子可信依赖类型及分配比

例进行初步的分析与说明。

首先，图 2 最右列为重组后业务流程，该流程共有 12 个构件算子，其中，代表强可信依赖类型的使用算子共有 8 个，占 66.7%，是强可信依赖密集型业务流程。

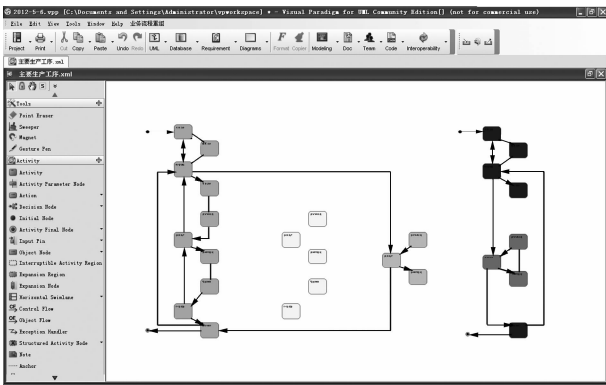


图 2 强可信依赖密集型业务流程重组实例

图 3 最右列为重组后业务流程，该流程共有 12 个构件算子，其中，代表半可信依赖类型的使用算子共有 7 个，占 58.3%，是半可信依赖密集型业务流程。

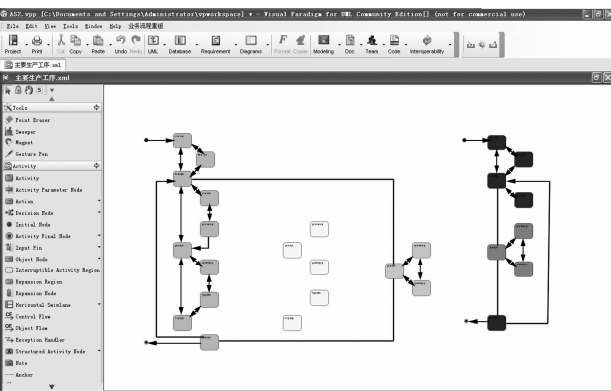


图 3 半可信依赖密集型业务流程重组实例

图 4 最右列为重组后业务流程，该流程共有 12 个构件算子，其中，代表弱可信依赖类型的算子共有 9 个，占 75%，是半可信依赖密集型业务流程。

通过对以上 3 种应用软件代表业务流程的测试，最终实验结果如图 5 所示。

通过对图 5 的分析可以看到，代表原始数据均值的曲线其可信值范围为[0.252 8, 0.723 3]，但是在将同组数据应用到 3 种类型的应用软件之后，所获得的度量可信值与其对应均值有显著差异，且不同类型应用软件反映出差异的趋势也不一样。

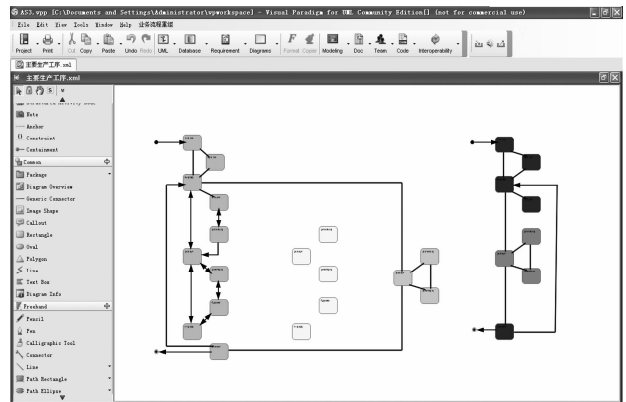


图 4 弱可信依赖密集型业务流程重组实例

首先，从共同特点来看，3 种类型的应用软件经度量后可信值普遍低于均值，即使是弱可信依赖密集型应用 AS3，也仅仅有不到 10 组数据达到或高于均值。这说明在考虑到构件间运算算子对可信值的依赖关系，并度量基于依赖关系影响的可信值后，所有应用软件可信性均比其原始数据均值有不同程度的降低。

其次，具体到每种类型的应用软件，其度量后的可信值降低程度不完全相同。其中半可信依赖密集型应用软件降低程度最大，这是因为随着具有半

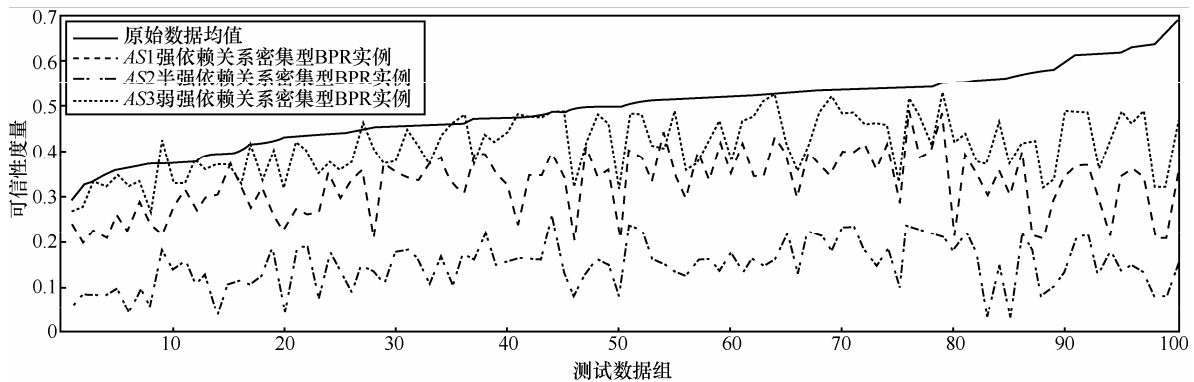


图 5 3 种业务流程重组实例可信性度量

可信依赖特征的调用算子比例的增加,在调用长度不小于 1 的调用算子链上关联的构件数量明显增加,而其可信性依赖的传递长度也在增加,所带来的影响造成了关联构件可信性的降低;而针对弱可信依赖密集型应用软件,其度量后的可信值降低程度最小,甚至少数情况下,其度量后的可信性略高于原始数据均值,这是因为具有弱可信依赖特征的算子种类有 5 种,其中部分算子是指参与该算子的构件之间具备容错和备份关系,例如并行算子,在处理关键业务的高性能计算机中,经常将相关构件设计为相互备份且并行运行模式,这样会提高构件群的整体可信性,这一点在该实验中有具体体现。而针对强可信依赖密集型应用软件,其度量后的可信值降低程度处于 3 种类型应用软件的中间程度,这是因为一方面随着具有强可信依赖特征的使用算子的增加,与使用算子链相关的构件可信性完全依赖于该链上可信性最低的构件,从而降低了构件群的整体可信性,但另一方面使用算子链不具备调用算子链的可信性传递消减关系,因此降低程度不如半可信依赖密集型应用软件大,这一点在该实验中也有具体体现。

再次,该实验还反映出一个显著特征,即无论是哪种依赖关系密集型的应用软件,其度量后的可信性均未体现出与其对应的原始数据均值保持相同的单调递增关系。究其原因,是因为原始数据均值仅仅是所有构件在该组数据中其独立可信性的平均值,并未考虑其相互之间的运算算子在可信性上产生的相互影响,另外对均值相对较大的构件可信性原始数据组,其度量后的可信性反而较低的原因是影响该业务流程的关键构件可信性发生了较大的程度降低。例如,第 1、2 组测试数据中,该 10 个构件的在未考虑构件算子情况下单独度量的可信性原始数据值为

No[1]:(0.371 0, 0.037 0, 0.515 0, 0.820 0, 0.424 0, 0.069 0, 0.152 0, 0.826 0, 0.199 0, 0.331 0), 原始数据均值为 0.292 9。

No[2]:(0.678 0, 0.579 0, 0.579 0, 0.779 0, 0.954 0, 0.965 0, 0.811 0, 0.806 0, 0.192 0, 0.236 0), 原始数据均值为 0.328 1。

经过考虑构件算子影响并度量后,其可信性分别是 0.239 1 和 0.194 9,即发生降低情况。经过分析可以看到虽然有 7 个构件可信性提高了,但是由于 $\{Op_3, Np_2, Op_e\}$ 3 个构件的可信性降低,最终通过强

可信依赖关系反映为应用软件整体可信性的降低。

同时,通过以上分析结果可知关键构件可信性是决定应用软件整体可信性的核心因素,上述实验是从正向角度由已知软件体系结构和单独构件可信性度量并识别影响软件整体可信性的核心构件,也可通过数据挖掘等方法由已知软件整体可信性及其演化过程识别核心构件并推导软件体系结构,这属于逆向软件工程的研究范畴,将在以后工作中展开研究。

6 结束语

workflow 系统在商务与科学计算中具有重要的作用,而为满足 workflow 系统中业务流程的需求变更,必须完成业务流程的重组任务。针对业务流程重组领域可信性度量与评估问题,本文重点从构件化 BPR 应用服务器软件体系结构的形式化研究入手,给出了 BPRAS 的软件体系结构代数模型,包括原构件及与其同构的用户层、服务层和资源层构件和基于 BPRAS 构件的 BPRAS 构件组合。在此基础上,提出一种 BPR 应用软件体系结构代数模型,并介绍了验证业务流程的形式化案例。正如本文对国内外相关工作的分析中指出的情况,当前已发表的文献资料多数是从上述软件体系结构的用户层和资源层的角度,对 workflow 系统用户层面向的服务提供方和服务请求方的可信关系进行研究,或对其资源层包含的资源自身的可信性进行研究,提出一系列的评估方法,以期提高 workflow 系统的工作效率和可信性。而针对其服务层可信性的研究工作还相对少。与当前基于 workflow 系统的业务流程可信性的研究工作相比,本文具有以下几点创新。

1) 以业务流程重组为核心,给出一种基于业务流程重组应用服务器的流程建模与可信性度量协作框架,利用该协作框架,可将业务流程静态设计时的语义验证与动态运行时的可信性度量关联起来,并通过提高核心构件可信性,以迭代方式提高业务流程重组的质量。

2) 以构件及其相互运算为核心,提出一种构成 BPRAS 的构件模型,基于该模型,以形式化的方法有效地表达通过构件运算完成执行业务流程的语义信息,从而为其后的度量方法奠定软件体系结构基础。

3) 以 BPRAS 支撑的业务流程组成的应用软件行为可信性度量为核心,提出一种基于多级范式的度量方法,并通过将构件算子划分为强可信、半可

信和弱可信3类可信算子,有效地降低了应用软件可信性形式化模型的复杂度,并为其可信性度量提供了一种计算方法。

4) 利用前期研发的可视化语义验证工具进行业务流程建模,通过对三类可信算子密集型的业务流程实例进行可信性度量实验,分析了在构件平均可信性提高过程中,不同类型的业务流程重组所形成的应用软件可信性演化特征。

下一步的研究工作将会进一步研究上述协作框架的关键技术,开发可信性度量工具并将其与可视化语义验证工具集成到一起,实现协作框架的集成工具。此外,还将研究基于多级范式的业务流程重组的约简方法,通过发现业务流程中具备高阶范式特征的子流程,将影响整体业务流程可信性构件的发现阶段从现在执行阶段提前到分析与设计阶段,从而降低业务流程重组的开发成本。

参考文献:

- [1] 陈火旺,王戟,董威.高可信软件工程技术[J].电子学报,2003,31(12A):1933-1938.
CHEN H W, WANG J, DONG W. High confidence software engineering technologies[J]. Acta Electronica Sinica, 2003,31(12A): 1933-1938.
- [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.
ZHAO H Q, SUN J. An algebraic model of service oriented trustworthy software architecture[J]. Chinese Journal of Computers, 2010, 33(5):890-899.
- [3] 张帆,江敏,吴怀广等.一种基于无干扰的软件动态行为可信性分析方法[J].计算机科学,2012,39(1):101-103.
ZHANG F, JIANG M, WU H G, et al. Approach for trust analysis of software dynamic behavior based on noninterference[J]. Computer Science, 2012, 39(1):101-103.
- [4] YUAN Y Y, HAN Q. A data mining based measurement method for software trustworthiness[J]. Chinese Journal of Electronics, 2012, 21(1):13-16.
- [5] HAN Q, QIAN Y S. VIMPM A tool to support BPR in integrated manufacturing[A]. Proceedings of TMEE2011[C]. Changchun, Chian, 2011. 910-913.
- [6] 罗海滨,范玉顺,吴澄. workflow 技术综述[J]. 软件学报,2000,11(7): 899-907.
LUO H B, FAN Y S, WU C. Overview of workflow technology[J]. Journal of Software, 2000, 11(7):899-907.
- [7] 史美林,杨光信,向勇等. WFMS: workflow 管理系统[J]. 计算机学报, 1999, 22(3):325-334.
SHI M L, YANG G X, XIANG Y, et al. WFMS: workflow management system[J]. Chinese Journal of Computers, 1999, 22(3):325-334.
- [8] CAO J W, ZHANG F, XU K, et al. Formal verification of temporal properties for reduced overhead in grid scientific workflows[J]. Journal of Computer Science and Technology, 2011, 26(6):1017-1030.
- [9] YU Z F, SHI W S. Queue waiting time aware dynamic workflow scheduling in multicluster environments[J]. Journal of Computer Science and Technology, 2010, 25(4):864-873.
- [10] 余阳,汤庸,潘茂林等. 时态工作流过程模型及其合理性验证[J]. 软件学报,2010,21(6):1233-1253.
YU Y, TANG Y, PAN M L, et al. Temporal workflow process model and its soundness verification[J]. Journal of Software,2010,21(6): 1233-1253.
- [11] VIRIYASITAVAT W, MARTIN A. A survey of trust in workflows and relevant contexts[J]. Communications Surveys & Tutorials, IEEE, 2011, (99):1-30.
- [12] TAO Q, CHANG H Y, YI Y, et al. A rotary chaotic PSO algorithm for trustworthy scheduling of a grid workflow[J]. Computers and Operations Research, 2011, 38(5): 824-836.
- [13] 胡春华,吴敏,刘国平. Web 服务工作流中基于信任关系的 QoS 调度[J]. 计算机学报, 2009, 32(1):42-52.
HU C H, WU M, LIU G P. QoS scheduling based on trust relationship in web service workflow[J]. Chinese Journal of Computers, 2009, 32(1):42-52.
- [14] SPITZ S, BÖCK P B, TÜCHELMANN Y. Trust-based resource allocation and evaluation of workflows in distributed computing environments[A]. ICSTE2010[C]. 2010.372-376.
- [15] 杨建华,谢高岗,李忠诚. 一种业务流自适应尽力采样方法[J]. 计算机研究与发展, 2006, 43(3):402-409.
YANG J H, XIE G G, LI Z C. A best-effort adaptive sampling method for flow-based traffic monitoring[J]. Journal of Computer Research and Development, 2006,43(3):402-409.
- [16] 李海波,战德臣,徐晓飞. 工作流业务规则语义的完整性验证技术[J]. 计算机研究与发展, 2009, 46(7):1143-1151.
LI H B, ZHAN D C, XU X F. Integration verification of workflow business rule semantic[J]. Journal of Computer Research and Development, 2009, 46(7):1143-1151.
- [17] 李斌,谢丰,陈钟. 一种面向业务的风险评估模型[J]. 计算机研究与发展,2011,48(9):1634-1642.
LI B, XIE F, CHEN Z. A business oriented risk assessment model[J]. Journal of Computer Research and Development, 2011, 48(9):1634-1642.
- [18] 王远,范玉顺.工作流时序约束模型分析与验证方法[J].软件学报, 2007, 18(9):2153-2161.
WANG Y, FAN Y S. A method of time constraint workflow model analysis and verification[J]. Journal of Software, 2007,18(9): 2153-2161.

作者简介:



韩强(1973-),男,回族,黑龙江阿城人,博士,北方民族大学讲师、硕士生导师,主要研究方向为可信软件、 workflow 技术。



袁玉宇(1971-),女,山西忻州人,博士,北京邮电大学教授、博士生导师,主要研究方向为可信软件、软件服务工程、软件质量及软件测试等。