

化学性服务组合

邓广宏^{1,2} 曹万华^{1,2} 李俊² 黄友澎² 程雄²

(哈尔滨工程大学计算机科学与技术学院 哈尔滨 150001)¹

(武汉数字工程研究所系统科研部 武汉 430074)²

摘要 为解决嵌入式系统服务化集成过程中服务的动态组合和重组等应用问题,借鉴化学合成、分子裂解、化学置换、分子复制等化学反应机制,设计了一套化学性服务组合模型。形式化描述了服务组合过程中服务合成、分解、替换、复制等服务行为及其相应的运算演化机制,并在此基础上结合 petri 网方法给出了服务演化模型,描述了多服务并行和依赖情况下的复合组合过程。最后以指控系统自适应服务组合应用需求为背景,实现了面向任务的服务组合系统,对化学性服务组合模型的应用效果和性能进行了验证和分析。实验结果表明,化学性服务组合模型满足了指控系统自适应服务组合的应用需求,增强了系统的灵活性、可重构性。

关键词 服务组合, 化学反应机制, 服务行为模型

中图分类号 TP301 **文献标识码** A

Chemical Service Composition

DENG Guang-hong^{1,2} CAO Wan-hua^{1,2} LI Jun² HUANG You-peng² CHEN Xiong²

(College of Computer Science and Technology, Harbin Engineering University, Harbin 150001, China)¹

(Department of Scientific Research on System, Wuhan Digital Engineering Institute, Wuhan 430074, China)²

Abstract For working out the problem of service dynamical composition and re-composition in service orient embedded system integration process, a chemical service composition model was proposed. By imitating the manner of chemical synthesis, molecular decomposition, chemical replacement and molecular reproduction in the chemical reactions, service behaviors in the composition process like composition, decomposition, replacement, replication and their operating and evolution mechanisms were formally described. Based on these service behaviors and the petri-net method, the service evolution model was proposed to describe the compound composition process with services reliance and parallel situations. And then, as an illustration for meeting the requirements of service self-adaptive composition in command and control system, an experimental system which is called chemical service composition system (CSCS) was designed and implemented to verify the availability and analyze the performance of the models and algorithms. The results of experiment show that the chemical service composition model satisfies the application request of service self-adaptive composition in command and control system, and effectively enhances the flexibility and reconstitution of embedded systems.

Keywords Service composition, Chemical reaction mechanism, Service behavior model

1 引言

面向服务架构和丰富的服务资源为实现 Internet 环境下松耦合、高效率的 Web 应用提供了基础。而在嵌入式系统中,受时效性需求、计算存储资源、系统拓扑构成等应用环境的限制,大多数应用都以功能模块、数据交互等方式在分布式环境下进行紧凑的嵌入式应用集成,不可能实现基于 Web 形式的服务化部署。对嵌入式应用进行服务化抽象,进而实现嵌入式服务的组合应用和动态重组,是提升嵌入式系统应用集成的灵活性、可重构性和健壮性的一条有效途径和迫需解决的问题。

服务抽象是将嵌入式应用的功能输出、数据交互等各种

应用属性进行服务化描述,然后结合 SOA 应用模式实现嵌入式应用的识别和访问,进而实现服务的组合应用和动态重组。服务组合过程包括服务发现^[1,2]、服务选择与匹配^[3,4]、服务动态组合^[5-8],以及服务替换^[9,10]等众多服务行为和技术细节。近几年来已有很多学者对这些服务行为展开了研究,但这些研究大都基于 Internet 等非嵌入式环境下 Web 服务的应用和组合,其应用过程及相关协议较复杂,多适用于企业级应用。在嵌入式应用系统中,受系统拓扑构成、时效性以及计算资源等应用环境限制,Web 模式中基于 Web 访问形式的服务组合使用了大量的 Web 计算和存储资源,同时其拓扑构成中对 Web 服务器的依赖也使其难以适应嵌入式环境的应用需求。因此, W. T. Tsai 等人提出了一种 RTSOA 架构以及实

到稿日期:2012-08-12 返修日期:2012-12-02 本文受国防预研项目(4010105010103,62101050101,513150802),船舶基金(09J3.4.1,10J3.7.2)资助。

邓广宏(1981-),男,博士生,工程师,主要研究方向为软件工程、嵌入式系统、指控系统,E-mail:stephenden@163.com;曹万华(1966-),男,研究员,博士生导师,主要研究方向为软件工程、指控系统。

现 RTSOA 所需解决的问题,并提出了一种实时性约束下的服务组合的优化选择算法^[11]。R. Joshi 等人则基于发布订阅模式和服务 QoS 提出了以数据为中心的实时 SOA 概念^[12],以实现嵌入式实时系统面向服务的应用集成。

在化学世界中,化学粒子在化学规律的主导下动态、自主地进行物质的合成和重构,这与服务组合应用极其相似。Mirko Viroli 等人借鉴化学反应思想,引入服务活性的概念,提出了基于元组空间的服务协同模型以及在此基础上的竞态服务的自组合机制^[7],但其在服务 QoS 层面设计了基于服务活跃度的选择和替换机制,没有对服务组合过程中的动态自主的行为机制进行深入分析。Claudia Di Napoli 等人则依据化学反应行为对服务组合进行了建模,将服务描述为具有相应 QoS 的节点和有向边的映射集,然后结合高阶化学语言 HOCL(The High Order Chemical Language)描述了工作流的组合过程^[13],但其对服务的描述过于粗糙,没有详细地展现服务组合的动态过程。本文从借鉴化学反应过程中动态、自主的行为机制的角度,形式化描述了服务组合的行为过程,为实现嵌入式系统中服务动态、自主的组合和重组奠定了基础。

2 化学反应与服务行为类比分析

将服务网络中各个服务视为化学反应中的原子和分子,服务组合过程类似于化学合成反应,二者都是将基本构成元素按一定的序列关联在一起形成新的合成元素过程。由此类推,服务重组与分子重排、服务替换与化学置换、服务分解与分子裂解等过程也很相似,服务行为和化学反应的类比图如图 1 所示。

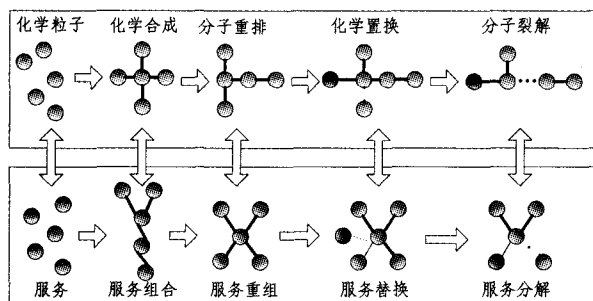


图 1 服务行为与化学反应过程类比图

更深一步分析,服务行为与化学反应机制在元素构成、关联关系、行为机制上都有对应的相似之处。服务和化学粒子构成了两个系统的基本元素;服务间的交互、协同或依赖关系以及化学粒子间的化学键力是这些元素间的关联关系;在服务关联关系和化学键的形成和断裂过程中衍生出了合成、重组、替换、分解等系列行为机制。服务行为和化学反应之间的对应关系如表 1 所列。

表 1 服务行为与化学反应机制的对应关系

组成元素	服务行为	化学反应机制
	服务	原子、分子
	服务合成	化学合成反应
	服务分解	分子裂解反应
行为机制	服务重组	分子重排反应
	服务复制	分子复制过程
	服务替换	化学置换反应
关联关系	服务交互、协同和依赖,数据关联,行为关联	共价键、离子键和金属键,电荷基团,氢键力,疏水基团等之间的相互作用力

通过类比分析服务行为与化学反应机制,借鉴化学反应机理,结合服务组合应用,本文提出了一种化学性服务组合模型,形式化描述了服务组合过程中服务键、服务合成、分裂、复制、替换等基本服务行为及其运算演化方式,以及在此基础上多服务并行和依赖等复杂情况下的复合组合模型,为构建动态自主的服务组合应用系统奠定了基础。

3 服务与服务键

3.1 服务

万维网协会(W3C)认为服务是一个软件系统,其公共接口和绑定方式通过 XML(Extensible Markup Language)来定义和描述,其他软件系统可以发现 Web 服务的定义,然后按照定义规定的方式与其交互^[14]。在嵌入式系统中,本文将服务视为基于某种上下文环境下的一组功能和属性的集合,基于这些描述属性,其它嵌入式应用可识别该应用并与其交互。嵌入式应用服务可以是实现某种功能的程序模块,也可以是满足某种应用需求的数据单元,或为二者的合集。

定义 1 服务是一个四元组: $Service = \{E, F, Q, I\}$ 。

(1) $E = \{R_1, R_2, \dots, R_n\}$, 表示服务应用环境集合,如系统平台、运行支撑库等。

(2) $F = \{D_1, D_2, \dots, D_m, O_1, O_2, \dots, O_n\}$, 表示服务功能集合,包括数据和操作两部分。其中 D_i 表示服务产生的数据输出, O_j 表示服务提供的操作接口。若服务是某个功能模块,则 O_j 表示服务的功能接口输出;若服务是某种数据单元,则 D_i 表示服务的数据输出信息。

(3) $Q = \{T, C, R, U, P\}$, 表示服务质量属性集合。其中 T 表示服务执行时间; C 表示服务一次执行所需的资源耗费; R 表示服务可靠性,标识服务持续正确执行的能力,采用服务平均无故障时间表示; U 表示服务可用性,采用服务的执行成功率来衡量; P 表示服务信誉,是服务应用方对服务的综合评价,采用平均服务需求满足度表示。

(4) $I = \{DI_1, DI_2, \dots, DI_m, FI_1, FI_2, \dots, FI_n\}$, 表示服务交互属性集合,其中 DI_i 表示服务交互过程中的数据输入, FI_k 表示交互过程中的事件的输入。

服务属性信息中既有数值型信息也有非数值型信息,对服务的非数值型属性信息基于描述逻辑^[4,8]规则进行表示。

3.2 服务键

服务键描述了服务组合体中各服务间的交互、协同、依赖等关联关系,如服务间的接口交互、数据交互等。

定义 2 服务 $A = \{E, F, Q, I\}$, $B = \{E, F, Q, I\}$, 若有 $A.F.D_1 \leftrightarrow B.I.DI_1$, $A.F.O_1 \leftrightarrow B.I.FI_1$ 等对应关联关系,则称 $\langle A.F.D_1, B.I.DI_1 \rangle$, $\langle A.F.O_1, B.I.FI_1 \rangle$ 为服务键,表示 A 的数据输出 D_1 与 B 的数据输入 DI_1 、 A 的事件输出 O_1 与 B 的事件输入 FI_1 等在服务合成体中建立了交互或依赖等关联关系。记为:

$$\langle A, B \rangle = \{ \langle A.F.D_1, B.I.DI_1 \rangle, \langle A.F.O_1, B.I.FI_1 \rangle \}$$

$\langle \langle A, B \rangle \rangle$ 记录了服务键中的元素集, $\langle \langle A, B \rangle \rangle = \langle \langle B, A \rangle \rangle$ 。

4 化学性服务组合模型

4.1 基本服务行为

4.1.1 服务合成

服务合成是多个服务在服务键的形成过程中形成功能聚

集的服务合成体的过程。

定义 3 若有服务 $A = \{E, F, Q, I\}, B = \{E, F, Q, I\}$, 服务 A 和 B 的服务合成记为:

$$S = A \Phi B = A \cup B - \{\langle A, B \rangle\}, \text{ 且 } \forall x \in S, x = T$$

其中, $A \cup B$ 是合成服务的主体部分。

$$A \cup B = \{A, E \cup B, E, A, F \cup B, F, A, Q \Delta B, Q, A, I \cup B, I\}$$

$\{\langle A, B \rangle\}$ 是合成服务的修正部分, 指合成过程中转换为服务键的属性集, 这部分属性不再包含在服务合成体的属性集中, 不对服务合成体产生外部作用。因此, 这部分属性通过修正从 S 的属性集中移除。

Δ 是服务的 QoS 合成运算符, 合成服务的 QoS 属性通过服务的 QoS 合成计算方法得出。计算方法如下:

(1) 并行服务

$$\begin{aligned} Q_1 \Delta Q_2 &= \{T, C, R, U, P\} \\ &= \{\max(Q_1.T, Q_2.T), Q_1.C + Q_2.C, \min(Q_1.R, \\ &\quad Q_2.R), \min(Q_1.U, Q_2.U), \min(Q_1.P, Q_2.P)\} \end{aligned}$$

(2) 串行服务

$$\begin{aligned} Q_1 \Delta Q_2 &= \{T, C, R, U, P\} \\ &= \{Q_1.T + Q_2.T, Q_1.C + Q_2.C, \min(Q_1.R, \\ &\quad Q_2.R), Q_1.U * Q_2.U, \min(Q_1.P, Q_2.P)\} \end{aligned}$$

在服务合成过程中, 存在多个服务要与一个服务进行合成的情况, 这时多个服务间的合成能否分开进行串行化处理, 即服务合成的分配律是否成立? 另外, 当多个服务顺序进行服务合成时, 能否根据应用需求变换服务合成的顺序? 这些演化规则都将影响服务合成过程成功与否。

定义 4 若 $\exists x \in A, F, x = T$, 当且仅当 $y = T, y \in A, E \cup A, I$, 即只有在服务 A 的环境属性或交互属性 y 成立时, A 的功能属性才成立, 则称服务 A 存在环境依赖或交互依赖。若 $\exists x \in A, F, x = T$ 当且仅当 $\langle a, b \rangle = T, \langle a, b \rangle \in \langle A, B \rangle$, 即只有当合成服务的服务键建立成功, A 的功能属性 x 才成立, 则称服务 A 依赖于服务 B , 记为 $A \mapsto B$ 。

若 $A \mapsto B$, 服务键 $\langle A, B \rangle$ 的形成会影响服务 A 的服务功能, 若服务 A 的某些功能失效, 则将会影响到后续的服务组合进程, 因此, 当多个服务进行组合操作时, $\langle A, B \rangle$ 的形成时机和顺序都会对服务合成产生影响。

定义 5 若服务 A, B 之间不存在相互依赖关系, 则称 A, B 相互独立。

定理 1 相互独立的服务间合成满足分配律。

证明: 即证明 3 个相互独立的服务 A, B, C 满足 $(A \cup B) \Phi C = A \Phi (C \cup B) \Phi C$ 。

根据服务合成的定义, 有:

$$(A \cup B) \Phi C = (A \cup B) \cup C - \{\langle A \cup B, C \rangle\}$$

式中, $\{\langle A \cup B, C \rangle\}$ 表示服务 A 和 B 与 C 的服务键集, 由于其相互独立, 服务键 $\langle A, C \rangle, \langle B, C \rangle$ 的形成序列不影响合成服务的功能, 可分解为 $\{\langle A, C \rangle, \langle B, C \rangle\}$, 因此, $(A \cup B) \Phi C = (A \cup B) \cup C - \{\langle A, C \rangle, \langle B, C \rangle\}$ 。

同时,

$$\begin{aligned} A \Phi (C \cup B) \Phi C &= (A \cup C - \{\langle A, C \rangle\}) \cup (B \cup C - \{\langle B, C \rangle\}) \\ &= (A \cup C) \cup (B \cup C) - \{\langle A, C \rangle, \langle B, C \rangle\} \\ &= A \cup B \cup C - \{\langle A, C \rangle, \langle B, C \rangle\} \end{aligned}$$

二者相等, 证毕。

定理 2 相互独立的服务间合成: 满足交换律。

证明: 两个服务的合成: $A \Phi B = A \cup B - \{\langle A, B \rangle\}, B \Phi A = B \cup A - \{\langle B, A \rangle\}, A \Phi B = B \Phi A$ 。

对于 3 个服务的合成, 选择 $A \Phi B \Phi C = A \Phi (C \Phi B)$ 来证明, 其它 5 种情况的证明同理。

根据服务合成的定义, 有:

$$A \Phi B \Phi C = (A \cup B - \{\langle A, B \rangle\}) \Phi C$$

由于其相互独立, 服务键 $\langle A, B \rangle$ 的形成时机和顺序不影响合成服务功能, 因此其可作为修正部分独立出来, 即 $(A \cup B - \{\langle A, B \rangle\}) \Phi C = (A \cup B) \Phi C - \{\langle A, B \rangle\}$, 再由服务合成的分配律得:

$$A \Phi B \Phi C = A \cup B \cup C - \{\langle A, B \rangle, \langle A, C \rangle, \langle B, C \rangle\}$$

同理:

$$\begin{aligned} A \Phi (C \Phi B) &= (A \cup C - \{\langle A, C \rangle\}) \Phi B \\ &= (A \cup C) \Phi B - \{\langle A, C \rangle\} \\ &= A \cup B \cup C - \{\langle A, B \rangle, \langle A, C \rangle, \langle B, C \rangle\} \end{aligned}$$

$A \Phi B \Phi C = A \Phi (C \Phi B)$, 证毕。

当服务间彼此不独立时, 服务键的形成时机和顺序会对服务合成过程中的服务依赖条件产生影响, 导致服务合成失败。因此, 当服务相互不独立时, 服务合成的分配律和交换律都不成立。

4.1.2 服务分解

服务分解是服务与服务合成体中的其它服务解除服务键关联关系、从服务合成体中独立出来的过程。

定义 6 对于 $S = A \Phi B$, 若服务体运行过程中满足 $A \notin S, B \notin S, \langle A, S - A \rangle = \emptyset, \langle B, S - B \rangle = \emptyset$, 则称服务合成体 S 分解为服务单元 A 和 B , 表示为: $S \triangleright A + B$, 其中:

$$A = S_A \cup \{\langle A, B \rangle\}_A; B = S_B \cup \{\langle A, B \rangle\}_B$$

S_A, S_B 表示服务合成体中属于服务单元 A, B 的属性集, $\{\langle A, B \rangle\}_A, \{\langle A, B \rangle\}_B$ 表示服务键元素集中各属于服务单元 A, B 的属性集。

定理 3 服务分解与服务合成操作是可逆的。

证明: 即证明通过合成与分解操作可以实现合成服务与独立服务之间信息无损的自由转换。

由服务合成和分解的定义可知, $A \Phi B = S \triangleright A + B$, 服务合成和分解可相互转换。其中, $A \Phi B = A \cup B - \{\langle A, B \rangle\}$, 合成过程中 A, B 的部分属性信息转换为 S 的服务键属性 $\{\langle A, B \rangle\}$, 其它部分转换为 S 的外部属性。同时, $S \triangleright A + B, A = S_A \cup \{\langle A, B \rangle\}_A, B = S_B \cup \{\langle A, B \rangle\}_B$ 。

分解过程将 S 的服务键和外部属性都外化为服务的外部属性。服务的合成与分解是服务键形成与断裂过程中属性信息的内外转换过程, 都没有信息损失。证毕。

4.1.3 服务替换

若服务合成体中出现服务失效或其它不适用情况, 则需选择新的服务来替换该服务。

定义 7 对于服务属性描述 x, y , 若逻辑推导 $x \Rightarrow y \wedge y \Rightarrow x$ 成立, 则称 x 与 y 逻辑等价, 记为 $x \Leftrightarrow y$ 。

定义 8 若服务 A 和 B 满足:

$$\begin{aligned} \forall x \in B, E, \exists y \in A, E, y \Leftrightarrow x \\ \forall x \in B, F, \exists y \in A, F, y \Leftrightarrow x \\ \forall x \in B, I, \exists y \in A, I, y \Leftrightarrow x \end{aligned}$$

则称服务 A 功能包含服务 B , 记为 $A \supseteq B$ 。

定义 9 若服务 A 和 B 满足:

$$A \supseteq B \wedge (\forall x \in B, Q, \exists y \in A, Q, y \Leftrightarrow x \wedge y > x)$$

则称服务 A 完全包含服务 B , 记为 $A \supseteq B$ 。

其中, $>$ 表示“优于”。针对不同的 QoS 属性, 采用不同的计算策略, 当值小为优时, 则小值优于大值; 当值大为优时, 则大值优于小值。

服务替换需要对服务进行可替换分析^[9,15], 然后通过分解, 合成过程形成新的服务体。受篇幅限制, 此处不作详细介绍, 仅给出服务可替换条件。

定义 10 若 $A \supseteq B$, 则服务 B 可被服务 A 替换。

4.1.4 服务复制

受拓扑网络以及安全性设计的限制, 一些服务有时候不能直接提供给应用方使用, 通过采用服务复制的策略将服务转移到一个可用的节点上可以让应用能够被正常访问和使用。服务复制是指在源节点以外的其它节点上提供源服务的应用功能的过程。

定义 11 若服务 A 和 B 满足 $A \supseteq B \wedge B \supseteq A$, 则称 A, B 为同构服务, 记为 $A \approx B$ 。

定义 12 若服务 A 和服务 B 同时满足 $A \supseteq B \wedge B \supseteq A$, 则称 A, B 为等价服务, 记为 $A \equiv B$ 。

在服务复制过程中, 服务首先探测到可复制的目标节点列表, 然后将其与应用方可访问的节点列表匹配, 选择一个双方都可访问的目标节点, 将嵌入式应用实体从源节点复制到目标节点, 并更新该服务的属性信息。初始情况下, 源服务 A 的属性信息会一一复制到目标服务 B 中, 因此 $A \equiv B$ 。服务执行过程中, 服务的可靠性和信誉度等 QoS 属性随服务的应用情况而产生调整, 存在 $A.Q \neq B.Q$ 的情况, 此时, $A \approx B$ 。因此, 若 $A \approx B$ 或 $A \equiv B$, 则可判断 A, B 在嵌入式系统属同源服务, 即二者皆为某类服务的复制, 服务功能相同, 仅存在服务质量上的差异。同源服务的记录和跟踪为服务的选择和替换过程节省服务寻找时间。

以上仅对服务复制的操作过程作了简要描述, 对服务复制的演化过程中的流程死锁^[16]、可信性保障^[17]等演化问题在此不作详述。

4.2 复合组合演化模型

在服务应用过程中, 服务组合往往不是单一行为。在组合策略和应用需求的驱动下, 服务合成、分裂、替换等服务行为在系统中执行各种形式的复合组合过程, 服务间也存在服务并行和服务依赖等复杂情况。通过分析服务行为过程中服务状态的变迁序列, 结合 petri 网描述方法给出了服务演化模型, 描述了多服务并行和依赖的复合组合演化过程。

定义 13 服务演化模型为一个五元组 $S = \{P, T, F, I, O\}$ 。

P : 有限库所集, $P = \{P_1, P_2, \dots, P_n\}$ 表示服务的状态集合。 P_1 : 游离态; P_2 : 准备态; P_3 : 挂起态; P_4 : 执行态; P_5 : 完成态。

T : 有限变迁集, $T = \{T_1, T_2, \dots, T_8\}$ 表示基本服务行为。 T_1 : 服务选择; T_2 : 服务执行; T_3 : 服务复制; T_4 : 服务失效; T_5 : 服务合成; T_6 : 服务依赖; T_7 : 服务分解; T_8 : 服务替换。

$F \subseteq (P \times T) \cup (T \times P)$ 是有向弧集合, 表示状态和行为的映射关系集。

I : 输入库所, 表示服务的初始状态。

O : 输出库所, 表示服务的终止状态。

基于服务演化模型, 服务分解、替换及合成的复合演化过

程如图 2 所示。Service 合成体中 Service2 服务失效, 因此 Service 分解: $Service \triangleright Service1 + Service2$, 然后 Service3 将 Service2 替换, 再进行合成过程 $Service = Service1 \oplus Service3$ 。服务复制及合成的复合演化过程如图 3 所示。Service1 访问受限, 因此将 Service1 复制到其它节点产生 Service3, 再与 Service2 进行合成。存在服务依赖的组合演化过程如图 4 所示。Service1 \triangleright Service2, 由于依赖条件不满足, Service1 在 P3 状态被挂起, Service2 执行过程中使得 Service1 依赖条件满足, 二者合成后得以继续执行。

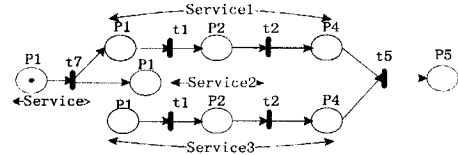


图 2 服务替换的复合演化过程

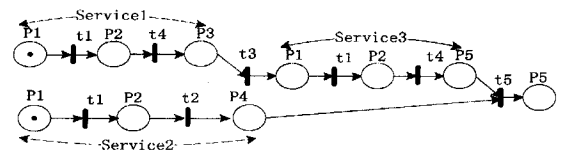


图 3 服务复制的复合演化过程

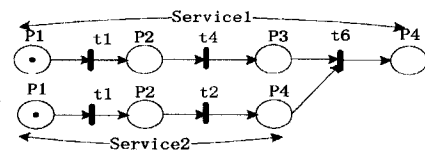


图 4 服务依赖的复合演化过程

一些更复杂的组合过程, 如多个服务混合依赖的服务组合等, 则是在以上基本组合形式的基础上进行组合设计实现的, 此处不一一列举。

定义 14 在服务合成、分解、替换、复制等基本服务行为及演化规则基础上的复合组合演化模型称为化学性服务组合模型。

5 实验与分析

在指控应用中, 根据指控任务动态自主地进行服务的组合和重组是快速构建指控业务流程、提升指控响应速度、实现指控智能化的重要基础。基于上节对化学性服务组合模型的分析, 本文以指控系统自适应组合应用需求为背景, 设计实现了一套面向任务的自适应服务组合系统, 系统应用化学性服务组合模型中的服务合成、服务分解、服务替换和服务复制等行为机制实现了面向任务需求描述的服务的自适应组合过程, 并在主频 2.8G、内存 2G 的 PC 机上进行了一个指控想定的实现, 如图 5 所示。

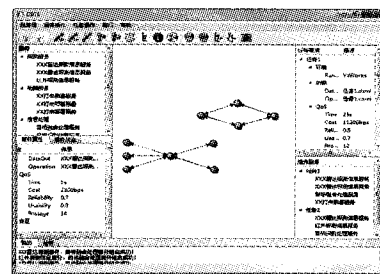


图 5 面向任务的化学性服务组合系统

在指控想定中,系统首先根据各类服务的应用接口,基于 Qt4.5 环境,在 PC 平台上仿真实现了 3 个态势监测服务、3 个信息处理服务和 4 个武器服务;然后将任务需求以服务的形式 $ServiceRequest = \{E, F, Q, I\}$ 描述任务的环境需求、功能需求、QoS 属性需求和交互需求,再与服务的 $\{E, F, Q, I\}$ 属性进行匹配,选择满足任务需求的服务;最后通过服务自组合算法进行服务的合成,形成服务组合体并从中优选一个作为组合结果。服务自组合算法如算法 1 所示。

算法 1 服务自组合算法

```

Input:
  ServiceRequest: 任务需求描述
  ServiceS: 服务集
Output:
  selectService: 满足服务需求的服务组合体
Steps:
1. ServiceSatisfied=NULL;
2. ServiceRelied=NULL;
3. ServiceIntegrated=NULL;
4. currentMatchDegree=0;
5. selectService=NULL;
6. For Service in ServiceS
7.  matchDegree=computeMatchDegree(ServiceRequest, Service);
8.  If matchDegree>threshold then
9.    ServiceRelied=search relied services in Service. E∪Service. I;
10.   ServiceSatisfied +=Service+ServiceRelied;
11.  End if
12. End for
13. For Service in ServiceSatisfied
14.  Invoke an agent to integrate Service;
15.   If Service. I ==NULL then
16.     ServiceIntegrated +=Service;
17.   elseDo
18.     find (Service,ServiceInteract);
19.     Service=Service∩ServiceInteract;
20.   While Service. I ==NULL
21.     ServiceIntegrated +=Service;
22.  End for
23. For Service in ServiceIntegrated
24.  matchDegree=computeMatchDegree(ServiceRequest, Service);
25.  If matchDegree > currentMatchDegree then
26.    currentMatchDegree=matchDegree;
27.    selectService=Service;
28.  End if
29.  ServiceRequest -=selectService;
30. End for
31. return selectService

```

服务自组合算法借鉴化学反应中随机碰撞反应原理,每个服务在组合过程中都作为一个主体,自主地与其它服务进行合成、替换等服务行为。算法为每个满足需求的服务启动一个自组合 agent,该 agent 根据服务的环境需求、交互条件等寻找服务键结合体,进行服务合成、替换等行为,形成服务组合体,最后根据需求匹配度从形成的多个组合体中选择最佳组合体作为组合结果。

指控想定的组合结果如图 6 所示。指控中心根据 Service1、Service2、Service3 等 3 个态势监测服务,通过信息处理服务 Service4 形成全局态势信息,然后根据全局态势信息的

告警提示,启动 Service5 和 Service6 两个武器服务进行功能组合,实现协同防卫的功能。

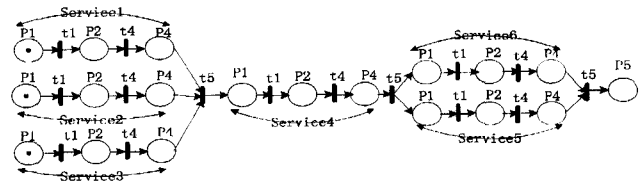


图 6 指控想定的组合演化流程

为评估系统对真实环境下批量服务组合的承载能力,本文研究了服务数增加情况下服务组合的执行时间。系统首先仿真实现了 50 个有 2~5 个交互接口的服务,然后执行服务组合过程,并在系统中插入时间戳,获得了服务筛选阶段、组合阶段及组合体选择阶段的耗时,时间值取 10 次运行的平均值,以提高数据的精确度。

实验结果如表 2 所列。从中可以看出,服务组合时间不是单纯地只与服务的数量有关,服务接口数、服务组合体的大小等对服务组合的执行时间都会产生影响。

表 2 服务组合时间随服务数量增加的仿真数据

序号	服务数量(个)	服务总接口数(个)	服务组合体大小(个)	组合平均执行时间(秒)			总计
				服务筛选时间	服务自组合时间	组合体选择时间	
1	10	42	3	1.026	2.812	0.785	4.623
2	20	81	6	1.274	3.323	0.764	5.361
3	30	115	10	1.406	3.659	0.789	5.854
4	40	152	11	1.463	3.833	0.782	6.078
5	50	173	12	1.489	3.915	0.783	6.187

服务组合时间和服务数量的关系如图 7 所示。可以看出,服务组合执行时间随服务数量的增加并非呈线性增长趋势,而且坡度趋于平缓,这主要是因为通过为每个服务自组合过程建立独立线程,让每个服务并行地进行自组合处理,缩短了服务自组合时间。

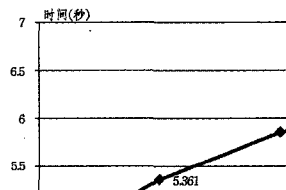


图 7 服务组合时间随服务数量变化图

以上实验表明,系统根据任务需求实现了服务的自适应组合,快速构建了指挥业务流程,满足了指控系统自适应服务组合的应用需求。同时,对批量服务组合也有较好的承载能力。但在实验中也发现,系统对服务动态自组合过程中出现的诸如服务组合冲突消解、组合方式的优选等问题考虑不足,影响了其效率和可靠性,这些问题还有待深入研究。

结束语 通过类比分析化学反应机制和服务组合行为的相似性,借鉴化学系统中动态自主的反应机理,设计了一套化学性服务组合模型;形式化描述了服务键、服务合成、服务分解、服务替换、服务复制等服务行为及其运算演化机制,并在此基础上结合 petri 网方法给出了服务演化模型,描述了服务并行和依赖等复杂情况下的复合组合过程;最后结合指控系统自适应组合的应用需求,实现了面向任务的化学性服务组

(下转第 56 页)

络结构的基本特征。并且,根据前述的化简原则,这两种化简结果无法区分优劣,因为它们化简后的特征环部分具有相同的特征点数目和相同的普通点数目。由此看见,基于特征环的网络化简方法的结果可能是不稳定的,可能存在不同的化简结果,并且这些结果无法比较优劣。

结束语 本文提出的基于特征环的复杂网络结构化简方法的特点在于,在被化简后的各个子网中,除了特征点之外,其它普通点均不能和外界有连接关系。文中提出的算法,其本质是扫描各个环路径,如果不是单纯特征环,则向外搜索包含普通点的闭合路径,直到构成复合特征环为止。之所以从最小的环路径开始扫描,是因为最小的环路径可以将网络划分为更小的部分。在复合特征环的扫描过程中,也应当遵循同样的规则,即复合特征环中应当包含尽可能少的节点。由于环路径选择顺序的不同,网络的化简结果也可能不同,并且可能无法比较化简结果的优劣,但这些不同的结果都是网络连接结构特征的反映。

参 考 文 献

[1] van der Aalst W M P. The Application of Petri Nets to Workflow Management [J]. The Journal of Circuits, Systems and Computers, 1998, 8(1): 21-66
 [2] van der Aalst W M P, Ter Hofstede A H M. YAWL: yet another

workflow language[J]. Information Systems, 2005, 30: 245-275
 [3] Ellis C A, Nutt G. Workflow: The Process Spectrum. In A. Sheeth, editor[C]//Proceedings of the NSF Workshop on Workflow and Process Automation in Information Systems. Athens, GA, May 1996: 140-145
 [4] Papazoglou M, Delis A, Bouguettaya A, et al. Class Library Support for Workflow Environments and Applications [J]. IEEE Transactions on Computers, 1997, 46: 673-686
 [5] van der Aalst W M P, Ter Hofstede A H M, Kiepuszewski B, et al. Workflow Patterns [J]. Distributed and Parallel Databases, 2003, 14(3): 5-51
 [6] Sadiq W, Orlowska M E. Analyzing process models using graph reduction techniques [J]. Information Systems, 2000, 25(2): 117-134
 [7] Vanhatalo J, Völzer H, Koehler J. The refined process structure tree [J]. Data & Knowledge Engineering, 2009, 68: 793-818
 [8] Wynn M T, Verbeek H M W, van der Aalst W M P, et al. Reduction rules for YAWL workflows with cancellation regions and OR-joins [J]. Information and Software Technology, 2009, 51(6): 1010-1020
 [9] Wynn M T, Verbeek H M W, van der Aalst W M P, et al. Soundness-preserving reduction rules for reset workflow nets [J]. Information Sciences, 2009, 179(6): 769-790

(上接第 20 页)

合系统,对组合模型的效果及性能进行了验证和分析。实验表明,化学性服务组合模型满足了指控系统自适应服务组合的应用需求,为指控流程组织的灵活性和智能化程度的提升奠定了基础。但在实验中也发现,诸如服务组合冲突消解、组合方式的优选等将影响系统的效率和可靠性,这些问题还需要更进一步的研究。

参 考 文 献

[1] He Li, Liu Shang. A Grid Service Discovery Method Based on Semantic Description [C]//2010 International Conference on Intelligent System Design and Engineering Application, Changsha, 2010: 40-44
 [2] 潘静,徐锋,吕建. 面向可信服务选取的基于声誉的推荐者发现方法 [J]. 软件学报, 2010, 21(2): 389-400
 [3] 王晓峻,毛莺池,钱国锋. 基于语义的 QoS 感知 Web 服务发现机制 [J]. 计算机科学, 2010, 37(8): 133-8
 [4] Jiang Zhi-xiong, Qian Le-qiu, et al. Dynamic Description Logic Based Services Semantic Matching [C] // 2008 Workshop on Knowledge Discovery and Data Mining, Adelaide SA, 2008: 229-234
 [5] Ibrahim A-O, Ahmed K. A Decentralized Self-Organizing Service Composition for Autonomic Entities [J]. ACM Transactions on Autonomous and Adaptive Systems, 2011, 6(1): 1-18
 [6] Poonguzhali S, Sunitha R, et al. Self-Healing in Dynamic Web Service Composition [J]. International Journal on Computer Science and Engineering, 2011, 3(5): 2055-2060

[7] Viroli M, Casadei M, et al. Spatial Coordination of Pervasive Services Through Chemical-Inspired Tuple Spaces [J]. ACM Transactions on Autonomous and Adaptive Systems, 2011, 6(2): 1-24
 [8] 刘思培,刘大有,齐红,等. 基于描述逻辑规则的语义 Web 服务组合 [J]. 计算机研究与发展, 2011, 48(5): 831-840
 [9] 刘莹,张一川,张斌,等. 基于行为效果的服务可替换性分析 [J]. 计算机研究与发展, 2010, 47(8): 1442-1449
 [10] 张教周,任洪敏,宗宇伟,等. 基于行为自动机的构件可替换性分析与验证 [J]. 软件学报, 2010, 21(11): 2768-2781
 [11] Tsai W T, Lee Y-H, Cao Zhi-bin, et al. RTSOA: Real-Time Service-Oriented Architecture [C] // Second IEEE International Symposium on Service-Oriented System Engineering, SOSE 2006. Shanghai, 2006: 49-56
 [12] Joshi R. Data-Oriented Architecture: A Loosely-Coupled Real-Time SOA [R]. white paper, Real-Time Innovations, 2007
 [13] Napoli C D, Giordano M, Németh Z, et al. Using chemical reactions to model service composition [C] // Second Int. Workshop on Self-Organizing Architectures ACM, New York, 2010: 43-50
 [14] W3C. Web Service Choreography Interface (WSCI) 1.0 [OL]. World Wide Web Consortium. <http://www.w3.org/TR/wsci>, 2002
 [15] 窦文生,吴国全,魏峻,等. 基于状态方面的 Web 服务动态替换 [J]. 计算机科学, 2009, 36(7): 97-101
 [16] 宋巍,马晓星,吕建. Web 服务组合动态演化的实例可迁移性 [J]. 计算机学报, 2009, 32(9): 1816-1831
 [17] 曾晋,孙海龙,刘旭东,等. 基于服务组合的可信软件动态演化机制 [J]. 软件学报, 2010, 21(2): 261-276