维普资讯 http://www.cqvip.com

一种基于分布构件装配元信息动态重构的表示方法*)

王克波　贾　焰　韩伟红　王志英
(国防科技大学计算机学院　长沙 410073)

摘　要　动态重构技术使得软件系统自身进行演进的同时，仍然能够提供服务。本文提出了一种基于构件装配元信息的动态重构操作的表示方法。该表示方法以Jeff提出的动态重构系统设定为基础，采用集合来表达构件应用中的构件、连接子和节点，为动态重构规划的生成以及等价性验证等提供了量化方法，并在基于CORBA构件模型的StarCCM分布构件平台中加以实现。

关键词　动态重构，构件，构件装配

A Representation of Dynamic Reconfiguration Based on Distributed Component Composition Metadata

WANG Ke-Bo　JIA Yan　HAN Wei-Hong　WANG Zhi-Ying
(Department of Computer Science, National University of Defense Technology, Changsha 410073)

Abstract　Dynamic Reconfiguration is the enable technology of non-stop 24x7 service, while the system evolves itself on line. An approach of representation of dynamic reconfiguration based on software component composition metadata is proposed, which builds itself on Jeff's viewpoint on system supporting dynamic reconfiguration, set is used as means of representation for components, connectors and machines. This approach is built in distributed component platform StarCCM, which conforms CORBA Component Specification.

Keywords　Dynamic reconfiguration, Component, Component composition

随着分布式技术向关键业务领域的快速拓展，分布式软件需要快速适应需求变化。动态重构技术为软构件及其体系结构的在线演化提供了有效的支持。软件在其生命周期运行阶段，有必要随需求而动，进行动态重构，其运行阶段由软件正常提供服务阶段与动态重构阶段相互交错而成。在软件的动态重构阶段，软件仍然能够提供正确的功能。

在运行阶段动态重构过程中，分布构件化软件需要在局部"停止"服务或者"微重启"[1]，以实现构件的增加替换等重构操作。不可避免地，分布构件化软件的性能与正常提供服务时相比会下降。重构操作序列越短，则重构开销相对较小。软件在进行重构时，多种不同的重构规划都能够达成相同的重构目标，庞杂的重构规划，需要保证正确性；不同开发者开发的重构规划，其等价性也需要验证；重构规划的重构操作数量不同，有些重构规划中可能隐含冗余的重构操作，导致不必要的性能损失。

构件化软件动态重构需要一种表示方法，一方面能够表述软件的动态重构，对软件重构规划的正确性、完备性等进行验证，以及为重构规划中冗余的剔除、验证重构规划等价性提供基础；另一方面，该表示方法其表达能力强，易于被开发者接受，并且容易机器实现，而且对重构规划的化简、等价判断等容易使用工具验证。

本文的方法将描述与构件组装元信息紧密结合起来，能够从构件包中直接抽取分布应用元信息，从而得到描述信息。通过描述信息能够对重构规划进行生成、验证、等价判断。

1　背景

下面介绍构件技术及动态重构。

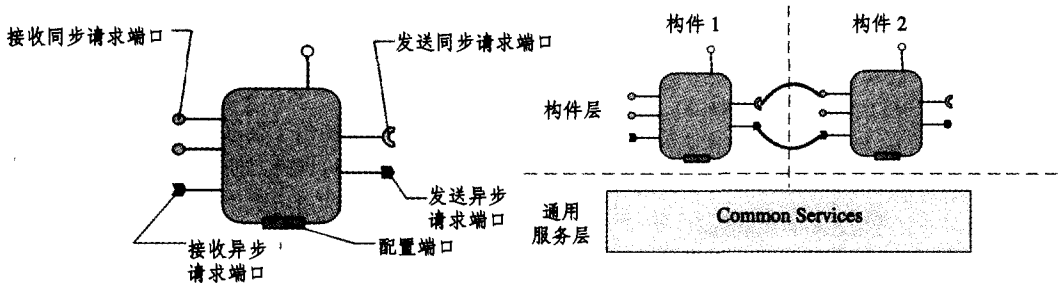

图 1　分布式构件及构件组装

典型的分布构件的应用，通常由分布在多个服务器节点　中的构件互相连接，构成构件组装(Assembly)，如图 1 所

*) Supported by the National Hi-Tech Research and Development Plam of China under Grant. No. 2004AA112020、2003AA115210、2003AA111020、2003AA115410(国家高技术研究发展计划(863))。王克波　博士生，主要研究领域为分布计算。

维普资讯 http://www.cqvip.com

示[7,8]。基于构件的系统由于具有对构件及构件之间的通信清晰描述的体系结构元信息，是天然适合应用动态重构技术的平台。构件包元信息是构件包的重要组成部分，包含了软件体系结构描述的三个要素，即构件、连接子和机器。

重构基本操作是分布构件平台能够识别由动态重构操作构成的动态重构规划(Reconfiguration Plan)。通常，基本动态重构操作一般包括五种：构件添加、删除，构件间连接的添加、删除，以及构件迁移(图 2)。这些重构操作对分布式系统的功能变更或扩展提供了良好的支持。

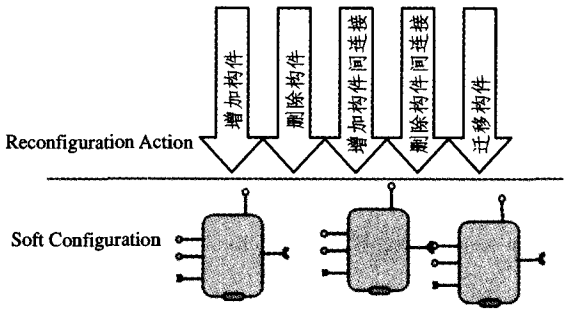

图 2　五种基本动态重构操作

支持动态重构的分布构件运行环境应该遵循 Jeff 为动态重构平台的设定[5]：

(1)重构应该以系统结构来表达；

(2)重构规范应该和应用的状态、协议和算法彼此独立；

(3)重构不会破坏系统的一致性状态；

(4)重构对应用的影响应该最小化。

以下的讨论均基于 Jeff 对动态重构平台能力的设定。我们实现了基于 CORBA 构件规范的分布构件平台 StarCCM[6]，能够满足 Jeff 设定。

2　相关研究

动态重构的研究主要从两个方向展开[1,4]：形式化方向和非形式化方向。形式化方向的研究重点在于对系统结构、语义、约束和配置变化的形式化描述以及相关性质的形式化验证。利用形式化的软件体系结构描述语言对系统的描述和分析技术，特别是对系统动态特性的描述和分析技术还远未成熟。

动态软件体系结构能够表示软件执行过程中对其结构的修改[1,2,4]。现有的形式化软件动态体系结构规范，可以分类为基于图(CHAM[10]、COMMUNITY[11])、基于逻辑(ZCL[12])、基于进程代数(Darwin[13]、Dynamic Wright[14])等。形式化方法的优点是能够满足语义要求，期望从理论上解决动态重构中一致性问题；其缺点也很突出，难度大，需要专家经验；涉及语义，相关技术仍在研究中，在实用中不成熟。

基于 AI 动态重构规划的方法[9]，将动态重构实现机制与规划结合起来，以感知、规划、动作为三个阶段，但往往只能用于一些简单程序[9]。

当前多数重构平台的实现都为非形式化方法。

3　体系结构重构过程及表示

构件级别的状态一致性由 StarCCM 平台提供支持，那么应用级别的一致性怎样保持？本文中的方法仅仅保持了应用静态结构的一致性。

为了方便表述，我们将动态重构的基本操作引入符号表述如下。

(1)添加构件：$\oplus C_i(N_1)$，表示构件 C_i 被添加到节点 N_1 上；

(2)删除构件：$\ominus C_i(N_1)$，表示构件 C_i 被从节点 N_1 上删除；

(3)添加构件间通信通路：$C_1(P_1)\leftrightarrow C_2(P_2)$，表示从构件 C_1 上的端口 P_1 到构件 C_2 上的端口 P_2 增加一条通讯通路；

(4)删除构件间通信通路：$C_1(P_1)\not\leftrightarrow C_2(P_2)$，表示将构件 C_1 上的端口 P_1 到构件 C_2 上的端口 P_2 之间的通讯通路删除；

(5)迁移构件：$C_i(N_1\Rightarrow N_2)$，表示将构件 C_i 从节点 N_1 迁移到节点 N_2 上；

(6)不操作：$\odot$，表示不做任何重构操作；

(7)重构操作的前驱后继关系：用符号“,”表示。

增加、删除连接重构操作不区分同步通讯和异步通讯。

分布构件化软件的配置可以用三元组$\langle S,L,D\rangle$表示，其中：

S 是构成分布式应用的构件组成的集合，即$\{C_i\}$；

L 是 S 中构件的所有出端口为定义域、所有入端口为值域构成映射，以$\rightarrow$表示映射，即$\{l\mid l\in f^{link}:C_{i_1}(P_{j_1})\rightarrow C_{i_2}(P_{j_2})\}$；

D 是以组成分布式应用构件集合为定义域、构件所有可部署节点为值域构成映射，即$\{d\mid d\in f^{dist}:C_i\rightarrow N_k\}$；

且 $i,j,k\in N$。

每个重构操作应用在系统上时，都会改变以上定义中三元组中的一个或者多个元素。

当两个分布构件化软件配置的三元组的三个元素相同时，称这两个配置相等。

当两个分布构件化软件配置的前两个元素相同时，称这两个配置功能等价。

采用三元组定义重新阐释每个重构操作。定义域 Domain、值域 Range(分别计算单个重构操作中的每个量，将重构操作分为动词和名词)。

重构发生前，构件化软件配置可以表示为$\langle S_1,L_1,D_1\rangle$，用 $Configuration_i$ 表示；重构之后构件化软件配置表示为：$\langle S_2,L_2,D_2\rangle$，用 $Configuration_d$ 表示。

则分布构件化软件从 $Configuration_i$ 通过实施重构规划 $ReconfPlan$ 到达配置 $Configuration_d$，即

$$Configuration_i \xrightarrow{ReconfPlan} Configuration_d$$

根据重构基本操作定义，软件配置 $Configuration_d$ 可以使用下列规则计算：

$\oplus C_i(N_1)$：则 $S_2=S_1+\{C_i\}$，$L_2=L_1$，$D_2=D_1+\{C_i\rightarrow N_1\}$；

$\ominus C_i(N_1)$：则 $S_2=S_1-\{C_i\}$，$L_2=L_1$，$D_2=D_1-\{C_i\rightarrow N_1 1\}$；

$C_1(P_1)\leftrightarrow C_2(P_2)$：则 $S_2=S_1$，$L_2=L_1+\{C_1(P_1)\rightarrow C_2(P_2)\}$，$D_2=D_1$；

$C_1(P_1)\not\leftrightarrow C_2(P_2)$：则 $S_2=S_1$，$L_2=L_1-\{C_1(P_1)\rightarrow C_2(P_2)\}$，$D_2=D_1$；

$C_i(N_1\Rightarrow N_2)$：则 $S_2=S_1$，$L_2=L_1$，$D_2=D_1-\{C_i\rightarrow N_1\}+\{C_i\rightarrow N_2\}$；

如果已知构件化软件配置$\langle S_1,L_1,D_1\rangle$和重构基本操作，新的软件配置也容易计算。

(下转封四)

维普资讯 http://www.cqvip.com

(上接第 279 页)

4 个例分析

研究动态重构的经典 Philosopher 问题[5](图 3),示范重、构规划等价判断、生成。

采用分布构件表示,三个哲学家构件 C_P,有三个端口 $C_P(P_1)$和 $C_P(P_2)$、$C_P(P_3)$,三个叉子构件 C_F,此构件具有一个端口 $C_F(P)$,一个观察者构件 C_B 及其端口 $C_B(P)$。设两个运行节点,分别为 N_1 和 N_2,而构件 C_B 运行在 N_2 上,其他分布构件运行在 N_1 上。

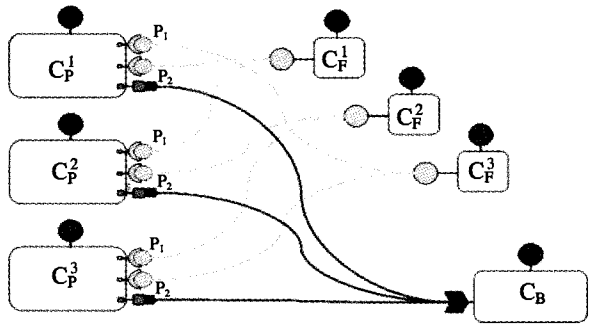

图 3 哲学家问题

哲学家问题采用前述的三元组进行描述,即为:

构件集合 S 为$\{C_P^1, C_P^2, C_P^3, C_F^1, C_F^2, C_F^3, C_B\}$;

如果→表示映射,其左边表示源,右边表示映射的目标,则 L 可以定义为如下映射构成的集合:

$\{C_P^1(P_1)\rightarrow C_F^1(P), C_P^1(P_2)\rightarrow C_F^3(P), C_P^2(P_1)\rightarrow C_F^1(P), C_P^2(P_2)\rightarrow C_F^2(P), C_P^3(P_1)\rightarrow C_F^2(P), C_P^3(P_2)\rightarrow C_F^3(P), C_P^1(P_3)\rightarrow C_B(P), C_P^2(P_3)\rightarrow C_B(P), C_P^3(P_3)\rightarrow C_B(P)\}$;

同上,以→表示映射,则 D 可以定义为如下映射所构成的集合:

$\{C_P^1\rightarrow N_1, C_P^2\rightarrow N_1, C_P^3\rightarrow N_1, C_F^1\rightarrow N_1, C_F^2\rightarrow N_1, C_F^3\rightarrow N_1, C_B\rightarrow N_2\}$;

如果将三个哲学家问题进行动态重构,变成四个哲学家问题,则需要添加一个哲学家构件 C_P^4 及一个叉子构件 C_F^4,假定它们会被部署在节点 N_2 上。

添加第四个哲学家构件和叉子构件,需要对原有构件之间的连接进行修改。容易将三个哲学家的情况推广到第四个,并且得到与前面类似的三元组。

根据前面提供的三元组计算规则,我们容易对其进行计算。

计算可得重构规划即重构操作序列为:

$\oplus C_P^4(N_2), \oplus C_F^4(N_2), C_P^1(P_2)\not\leftrightarrow C_F^3(P), C_P^4(P_3)\leftrightarrow C_B(P), C_P^1(P_2)\leftrightarrow C_F^4(P), C_P^1(P_1)\leftrightarrow C_F^1(P)$

总结 上述工作基于支持动态重构的开源 CORBA 构件平台 StarCCM,并且在 StarCCM 上得到了验证[6]。

本文提出了一种基于构件装配元信息的动态重构表示方法,本质上是一种基于图的方法。本方法的优点是容易从构件包元信息抽取得到描述信息,容易进行重构规划的计算以及等价性验证;缺点是没有引入语义信息,不能从理论上实现完全的一致性,一部分一致性实施工作需要开发者实现。

参 考 文 献

1 窦蕾. 面向构件的复杂软件系统中动态配置技术的研究:[博士学位论文]. 国防科技大学, 2005
2 Bradbury J S, Cordy J R, Dingel J, et al. A survey of self-management in dynamic software architecture specifications. In: WOSS'04: Proceedings of the 1st ACM SIGSOFT workshop on Self-managed systems, 2004. 28~33
3 Castaldi M. Dynamic Reconfiguration of Component Based Application:[PhD Thesis]. University of L'Aquila, 2004
4 Dou L, Wu Quanyuan, Jia Yan, et al. Dynamic Reconfiguration Platform Based on Distributed Component Technology CCM. Advanced Parallel Programming Technologies. In: 5th International Workshop, APPT ,2003
5 Kramer J, Magee J. The Evolving Philosophers Problem: Dynamic Change Management. IEEE Transactions on Software Engineering, 1990,16(11)
6 StarCCM, an Open Source CORBA Component Platform. http://starccm.sourceforge.net
7 CORBA Component Model Specification. Object Management Group, 2002. http://www.omg.org
8 Enterprise JavaBeans Specification. JCP, 2001. http://www.jcp.org
9 Arshad N, Heimbigner D, Wolf A L. Deployment and Dynamic Reconfiguration Planning for Distributed Software Systems. ICTAI '03. In: Proceedings of the 15th IEEE International Conference on Tools with Artificial Intelligence, IEEE Computer Society, 2003
10 Wermelinger M. A simple description language for dynamic architectures. In: Proc. of the Int Software Architecture Workshop, 1998. 159~162
11 Wermelinger M, Lopes A, Fiadeiro J L. A graph based architectural (re)configuration language. In: Proc. of the Joint European Software Engineering Conference and Symp on the Foundations of Software Engineering, 21~32
12 DePaula V C C. ZCL: A Formal Framework for Specifying Dynamic Software Architectures:[PhD thesis]. Federal University of Pernambuco, 1999
13 Magee J, Kramer J. Dynamic structure in software architectures. In: Proc. of the ACMSIGSOFT Symp. on Foundations of Software Engineering, 1996. 3~14
14 Allen R, Douence R, Garlan D. Specifying and analyzing dynamic software architectures. Lecture Notes in Computer Science, 1998,1382

计算机科学

(1974 年 1 月创刊)
第 33 卷第 11 期 (月刊)
2006 年 11 月 25 日出版

国际标准连续出版物号 ISSN 1002-137X
国内统一连续物出版号 CN50—1075/TP

定价:30.00 元 国外定价:5 美元
邮发代号:78—68
发行范围:国内外公开

主管单位:国 家 科 学 技 术
主办单位:国 家 科 技 部 西 南 信 息
编辑出版:《计 算 机 科 学》 杂
重庆市渝中区胜利路 132 号 邮政编码:
电话:(023) 63500828 E-mail:jsjkx@
网址:www.jsjkx.com
社 长:牟炳林
总 编:彭 丹
主 编:朱宗元
主编助理:徐书令
印 刷 者:重 庆 科 情 印 务 有 限 公 司
总发行处:重 庆 市 邮 政 局
订 购 处:全 国 各 地 邮 政 局
国外总发行:中国国际图书贸易总公司 (北京 399 信箱)
国外代号:6210—MO

P-2817
21.33(11)