文章编号 1000-5013(2006)03-0325-04

一种面向工程的构件式软件体系结构模型

李向阳 连小绮

(华侨大学工商管理学院, 福建 泉州 362021)

摘要 从软件体系结构服务于软件工程出发, 总结出6种基本模型元素, 建立一种新的构件式体系结构模型——EOCSAM模型, 给出它的形式语法和语义. 与其他模型不同的是, 它将构件的接口、连接模式、性能等作为体系结构的第1类模型元素, 并给出计算法则. 它能更好地实现从应用需求到代码实现之间的逐步过渡映射, 能够从功能、结构和任意多种性能的侧面对系统进行逐层跟踪分析, 并可作为软件开发全过程的协同工作框架.

关键词 软件体系结构, 形式化模型, 软件构件, EOCSAM模型, 构件式体系结构

中图分类号 TP 311 文献标识码 A

不同于现有的其他模型和描述语言[1~3], 本文介绍的面向工程的软件构件体系结构模型(简称EOCSAM模型). 它提炼出6类基本体系结构模型元素, 综合反映软件工程过程中的各方面的关键属性, 可对系统的功能、结构及自定义和预定义的各种性能指标进行逐层细化跟踪分析, 还起到一个支持全软件生命周期、各方面相关人员的协同工作框架作用.

1 模型目标

现有的体系结构模型主要从静态的、单纯组织结构和最终结果的角度来描述构件. 它们都不支持模型性能可跟踪的逐步求精, 没有考虑与软件工程过程的结合[4,5]. 软件体系结构的作用贯穿于软件的整个生命周期. 它反映系统的业务需求, 是项目管理、需求分析、详细设计、编程实现等众多相关人员理解系统、相互交流、进行功能和性能分析的依据. 在系统分析阶段和总体设计阶段, 它起到系统的抽象模型的作用; 在开发阶段, 它是早期设计和实施的蓝图, 是业务需求与最终实现代码之间的桥梁; 在运行维护阶段, 它又是配置和修改的重要资料. EOCSAM模型就是基于这样的认识, 从软件工程全过程出发建立的一个综合模型.

2 EOCSAM体系结构模型的语法和语义

EOCSAM模型由构件(Component)、构件构成模式(Structural Pattern)、接口(Interface)/接口函数(Interface Function)、输入信息项(Input)/输出信息项(Output)、数据类型(DateType)、性能(Performance)等6种项(Term)构成, 它们是构成模型的基本元素.

2.1 基本项的定义和运算法则

2.1.1 *数据类型* 数据类型是一集合, 它用以描述信息的组成结构及取值范围. 名称作为它的唯一标志, 类型相互之间的关系有相等关系和相容关系. 即对于两个数据类型 DT_1 和 DT_2, 当它们具有相同的结构, 且 $DT_1 \subseteq DT_2$ 时, 称为 DT_1 与 DT_2 兼容. 类型运算的组合, $DT = DT_1 \cup DT_2$ 组合关系, 形成新的类型. 在程序设计理论的类型理论中, 它们已有成熟的理论和实现方法, 可直接借用.

收稿日期 2005-10-06

作者简介 李向阳(1971-), 男, 讲师, 博士研究生, 主要从知识处理与知识工程及软件体系结构的研究. E-mail: lxy0601@sohu.com

基金项目 华侨大学科研基金资助项目(04HZR18)

2.1.2 性能 性能是一类特殊的数据类型,它用以描述数据流动吞吐量、时延、表示效果等侧面.性能之间也有一定的运算关系.不同类型的性能指标在不同组合方式下有不同的运算规则,如吞吐量在并运算下为和,时延在并运算下取最大值;在串联运算下,吞吐量取最小值,时延则为两者之和.还有一些性能指标和计算规则可具体的应用而设定.

2.1.3 输入与输出信息项 它们是接口函数的构成元素,输入信息项在抽象语法中表示为一个三元组(InputID, DataType, P),它们分别代表输入项标志、数据类型和输入信息流的性能.由于一个接口函数通常有多个输入信息项,输入项标志用以表示它们之间的区别.输出信息项是一个二元组(DataType, P),一个接口函数有且只有一个输出信息项.

2.1.4 接口函数 接口函数为构件提供信息流通道,将输入信息流转化为输出信息流.在抽象语法中表示为一个三元组,即 IF=((Input$_1$, …, Input$_j$, …, Input$_m$), Output, P),也可表示为 IF=(Input$_1$, …, Input$_j$, …, Input$_m$) $\xrightarrow{P}$ Output.其中的 P 表示该接口函数在处理信息转换过程中的相关性能.

2.2 接口函数之间的连接运算

接口函数之间的连接运算是实现构件之间相互连接的基础,构件之间的结构关系就是通过构件的不同子构件中接口函数之间的连接而实现的.当多个接口函数之间在输入/输出信息项相容的条件时,可以相互连接,形成新的接口函数.

2.2.1 两个接口函数之间的连接和表示 设有任意两个接口函数 IF_a 和 IF_b,即 IF_a=((Input$_{a1}$, …, Input$_{aj}$, …, Input$_{am}$), Output$_a$, P$_a$); IF_b=((Input$_{b1}$, …, Input$_{bj}$, …, Input$_{bn}$), Output$_b$, P$_b$). 当 IF_a 的输出项类型与 IF_b 的某个输入项相容时,IF_b 的这个输入项可以用 IF_a 接口函数取代.例如,当 I$_a$. Output$_a$ 的类型与 IF_b. Input$_{b1}$ 的类型相容时,后者可以用前者替换,实现两个接口函数的连接,形成一复合接口函数.记为 IF_a, $IF_b \Rightarrow IF_a \xrightarrow{Input_{b1}} IF_b$, if IF_a. Output$_a$. DataType $\subseteq IF_b$. Input$_{b1}$. DataType 连接相关的输入信息项也只用序号表示,即 $IF_a \xrightarrow{Input_{b1}} IF_b = IF_a \xrightarrow{1} IF_b$.两个接口函数项通过输出与输入进行串接之后,整体上它们仍是一接口函数项.因而上式可记为 $IF_c = IF_a \xrightarrow{1} IF_b$=((IF$_a$. Input$_{a1}$, …, IF$_a$. Input$_{aj}$, …, IF$_a$. Input$_{am}$, IF$_b$. Input$_{b2}$, …, IF$_b$. Input$_{bj}$, …, IF$_b$. Input$_{bn}$), IF$_b$. Output$_b$), IF$_c$. P$_c$) 其中 IF$_c$ 的输入信息项是 IF_a 输入信息项序列插入到 IF_b 中 Input$_1$ 位置之后的新序列,输出项为 IF$_b$. Output$_b$,性能则照相应的性能项的串行运算规则得到.

2.2.2 多个接口函数之间的连接和表示 多个接口函数之间的有两种基本连接模式.(1) 多个接口函数的输出分别并行地与某一个接口函数的各个输入项相连接,形成多流汇聚的模式,实现多个信息流通道的汇集.(2) 多个函数项一一首尾连接,形成多流串通的模式,实现多个信息流通道的连通.每种连接模式实际上由多次连接形成,每次连接都会消去一个输入项.对于多流汇聚连接模式,当接口函数项 IF$_{a1}$, IF$_{a2}$, …IF$_{an}$的输出项分别与 IF_b 的多个输入项 Input$_{b1}$, Input$_{b2}$, …, Input$_{bn}$相匹配并同时连接时,得到一新接口函数项,设为 IF$_c$,如图 1(a)所示.其中 m_1, …, m_i, …, m_n 表示各个函数项的输入项数.对于多流串通的情况,当有 n 个函数项 IF$_{a1}$, IF$_{a2}$, …, IF$_{an}$的输出项分别用前一项的输出与后一项的输出顺序地依据特定输入项首尾串行连接时,所得连接结果函数项的输入为各个函数项在去除被消除项之后并集,输出项为最后一个函数项 IF$_n$ 的输出项,性能也按相应的性能计算法则得到.设得到的新接口函数项为 I$_c$,如图 1(b)所示.其中 ki 表示第 i 个函数项中被第 $i-1$ 个函数项的输出取代的输入项序号.在实际应用中,多个函数之间的连接常常会是以上两种基本连接模式的混合.混合后相关输入输出项及相关性能的计算,可在以上法则的基础上自然推广得到.

2.3 构件式体系结构模型的形式语法

以下先给出表示构件体系结构模型中元素及相互之间的关系的符号系统,然后再逐条对它们进行解释.(1) C=(I, S, P).(2) I={IF$_1$, …, IF$_i$, …, IF$_n$}.(3) IF$_i$=((Input$_1$, …, Input$_j$, …, Input$_m$), Output, P).(4) Input$_j$=(InputID, DataType, P).(5) Output=(DataType, P).(6) S={Schema$_{IF_1}$, …, Schema$_{IF_i}$, …, Schema$_{IF_n}$}.(7) Schema$_{IF_i}$=C$_a$. IF$_m$|(C$_a$. IF$_i$ $\xrightarrow{k}$ C$_b$. IF$_j$)*.(8) P={P$_1$, …, P$_i$, …, P$_m$}.第1项表示构件 C 是一个(I, S, P)三元组合,其中 I 表示接口,由接口函数集组成;S 为结构模式,用于描述各个接口函数的来源和组成结构;P 表示该构件的性能集合.特定构件 C 中的接口 I 可通过符号“.”连接,表示为 C. I.同样 S 和 P 也可表示为 C. S 和 C. P.本文中所有语法元素的上下层之间都通过“.”连

$IF_c=(IF_{a_1}\rightarrow_1 IF_b, IF_{a_2}\rightarrow 2IF_b, \ldots, IF_{a_i}\rightarrow iIF_b, \ldots, IF_{a_n}\rightarrow nIF_b)$

$=((IF_{a_1}.Input_1, \ldots, IF_{a1}.Input_j, \ldots, IF_{a1}.Input_{m1},$

$\vdots$

$IF_{a_i}.Input_1, \ldots, IF_{a_i}.Input_j, \ldots, IF_{a_i}.Input_{mi},$

$\vdots$

$IF_{a_n}.Input_1, \ldots, IF_{a_n}.Input_j, \ldots, IF_{a_n}.Input_{mn},$

$), IF_b.Output), IF_{c.P})$

(a) 多流汇聚

$IF_c=(IF_{a_1}\rightarrow_1 IF_b, IF_{a_2}\rightarrow 2IF_b, \ldots, IF_{a_i}\rightarrow iIF_b, \ldots, IF_{a_n}\rightarrow nIF_b)$

$=((IF_{a_1}.Input_1, \ldots, IF_{a1}.Input_j, \ldots, IF_{a1}.Input_{m1},$

$\vdots$

$IF_{a_i}.Input_1, \ldots, IF_{a_i}.Input_j, \ldots, IF_{a_i}.Input_{mi},$

$\vdots$

$IF_{a_n}.Input_1, \ldots, IF_{a_n}.Input_j, \ldots, IF_{a_n}.Input_{mn},$

$), IF_b.Output), IF_{c.P})$

$,j\neq ki\ i=2,3,\ldots,n$

(b) 多流串连

图 1　接口连接模式

接. 第 2 项说明构件的接口由函数项集合构成, 每个构件具有特定的接口函数集, 其中的每个元素都是唯一的, 因而对于特定构件 C 来说, 其中的某个接口函数项 IF_1 可记为 C. I. IF_1, 也可以直接记为 C. IF_1. 接口通过接口函数为构件提供与外部交互的通道. 第 3 项说明了接口函数项由输入信息项列表、输出信息项及相应的处理性能 3 部分构成. 它又可以表示为$(Input_1, \ldots, Input_j, \ldots, Input_m)\overset{\rightarrow}{}_P Output$. 可解释为以 m 个信息项作为输入, 以 P 的性能进行处理后, 得到输出项 Output. 第 4, 5 项分别提供了对接口函数输出/输出信息项的定义. 第 6 项定义了构件的构成模式 S. 它是一个由模式项构成的集合, 每个组合函数项对应一个模式项, 形成一一对应关系. 第 7 项描述模式项的实际构成. 每个模式项描述了本构件中一个组合接口函数项, 如节 2. 2 所述的生成结构. 它说明一个函数项是如何由下层子构件的接口函数的相互连接而构成的. 第 8 项性能 P 是由多个性能项构成的集合, 每个性能项分别反映用户、设计者或其他相关人员关心的一个性能指标. 性能因子与构件、接口函数和输入/输出项相关联, 并随着构件和接口函数的结构组合而形成计算关系. 这种组合过程形成由粗到细, 可逐步追踪的性能网络.

2. 4　构件的组合运算与结构模式的描述

2. 4. 1　*基本构件与组合构件*　依据构件的结构特性不同, 可以分为基本构件和组合构件两大类. 当一个构件的每个接口函数项都不再由其他构件的接口函数组合时, 称该构件为基本构件; 否则, 称为组合构件. 当构件 C 为基本构件时, C. S 为空集. 对于组合构件, 其模式项集 C. S 非空, 每一项描述一个接口函数项由子构件接口函数组合生成的路径, 它们综合起来就形成该构件的体系结构描述. 构件是体系结构的构成要素, 同时也是体系结构的承载体. 构件之间相互引用或嵌套, 形成一个递归的、可逐层追踪细化和分析的体系结构模型. 将整体的系统视为一个最顶层构件, 则整个软件系统是具有唯一根结点, 一个自顶向下、逐层细化的多叉构件树; 每个作为结点的构件中都包含局部的小模式(图 2).

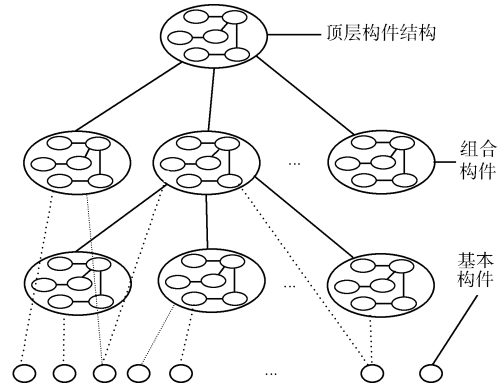

图 2　EOCSAM 体系结构树示意图

2. 4. 2　*构件内部组合的描述*　在构件抽象语法中, 基于接口函数连接运算规则实现由子构件组合而成新构件的组合结构描述提供了基础. 为使得模型更清楚地反映构件组合的现实, 在由多个子构件组合成一新构件时, 通常要引入一新封装对象. 本模型中将此封装对象也视为一构件, 令其为 C_0. 设有两个构件 C_a 和 C_b, 不妨设它们都为基本构件. $C_a=(I_a, S_a, P_a)=(\{IF_{a1}, IF_{a2}, IF_{a3}\}, \{\}, P_a)$ $C_b=(I_b, S_b, P_b)=(\{IF_{b1}, IF_{b2}\}, \{\}, P_b)$ $C_0=(I_0, S_0, P_0)=(\{IF_{01}, IF_{02}\}, \{\}, P_0)$ $C=(I_c, S_c, P_c)=(\{IF_1, IF_2\}, \{s_1, s_2\}, P)$. 其中, 模式项 s_1 描述了 C. IF_1 的内部结构, 同时也说明了在完成 IF_1 功能时各个构件的协作情况. s_1 的符号表示为 $s_1=(C.IF_1, (C_a.IF_{a1}\overset{\rightarrow}{}_1 C_b.IF_{b1}, C_a.IF_{a3}\overset{\rightarrow}{}_2 C_b.IF_{b1}, C_b.IF_1\overset{\rightarrow}{}_1 C_0.IF_{01}))$. 该关系也可以用图形化方式表示(图 3). 实际应用的 S 集合中经常包含多个模式项, 每个图元描述一个组合函数的结构.

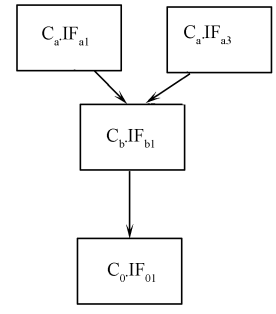

图 3　接口函数的构成示例

3 模型对软件工程过程的支持

EOCSAM 模型可以与软件工程过程很好地结合,应用于其生命周期的各个阶段. 在需求分析阶段,用户的需求可按类别组织后可与构件总体功能对应,细化的功能条目可与接口函数项功能相对应,这为构件的设计提供依据. 对构件和接口函数的解释随着所属的构件形成一个逐层细化的功能分解树,它是系统的需求分析结果的体现. 该模型为需求分析人员与体系结构设计人员之间工作的一个接合点,作为需求与代码之间的桥梁,对接口函数项和构件解释的结果,即"功能"起到第1个桥墩的作用.

模型中的接口函数项是对功能的符号化和形式化表示. 构件的接口函数的连接形成了具体功能项的逐层细化、以接口函数为结点的接口关联图,它是体系结构这一桥梁中跨接功能与设计之间的一个桥墩,也是体系结构设计人员与详细设计人员之间的工作接合点. 性能反映了基本功能之外的多元要求. 不同的项目有不同的性能要求,通过在模型元素中嵌入性能项,它随着模型结构的细化和组合实现各项性能指标由表到里,由需求到设计和实现以及它们在整个系统结构中的可跟踪路径. EOCSAM 模型的性能是一个开放集,它可随项目的要求而定义. 每一种性能指标都随构件由上到下的层次展开,形成一个相对独立的性能支持树,下层构件集的该性能指标值按一定的计算关系支持上层的值,便于对系统性能进行跟踪分析和模拟、找出瓶颈.

4 结束语

本文用形式化的方式描述了 EOCSAM 模型描述的构件体系结构,该模型将功能与功能分解,接口与接口关联,性能与性能支持树等集成在一个统一结构的符号系统中,易于通过不同视图对模型的特性进行观察和分析. 形式化的描述保证了理论上的可实现性,而它在实践中与其他软件工程模型和工具如 UML 及支持工具的结合,还值得进一步的研究,也是下一步工作的内容之一.

参 考 文 献

1 Medvidovic N, Taylor R N. A classification and comparison framework for software architecture description languages [J]. IEEE Transaction on Software Engineering, 2000, 26(1): 483~491

2 李 阳, 吴朝晖. 网络构件软件体系模型并行算法研究[J]. 浙江大学学报(工学版), 2004, 38(4): 392~396

3 孙昌爱, 金茂忠, 刘 超, 等. 软件体系结构研究综述[J]. 软件学报, 2002, (7): 1 228~1 237

4 骆华骏, 唐雅松, 郑建丹, 等. 可视化体系结构描述语言 XYZ/ADL[J]. 软件学报, 2000, (11): 1 024~1 029

5 赵会群, 王国仁, 高 远, 等. 软件体系结构抽象模型[J]. 计算机学报, 2002, (7): 730~736

An Engineering Oriented Component Software Architecture Model

Li Xiangyang Lian Xiaoqi

(College of Information Science and Engineering, Huaqiao University, 362021, Quanzhou, China)

Abstract Based on the key point that the software architecture is for the software engineering, this paper summarized six basic model elements, constructs a new component software architecture model named as EOCSAM and gave a formal grammar and semantic. In contrast with the other models, it uses the port, connection schema and performance as the first model element, it can map from the application requirement to final code smoothly, the function, structure and various performance targets can be conveniently traced and analyzed, it can also be used as a cooperating working framework among all roles in the whole project process.

Keywords software architecture, formal model, software component, EOCSAM model, componentual architecture