

# 面向体系结构的构件接口模型及其形式化规约

任洪敏<sup>1</sup>, 张敬周<sup>2</sup>, 钱乐秋<sup>2</sup>

(1. 上海海事大学计算机系, 上海 200135; 2. 复旦大学计算机科学系软件工程实验室, 上海 200433)

**摘要:** 借鉴软件体系结构的思想, 提出了面向体系结构的构件接口模型, 它既能表达体系结构设计的高层抽象构件, 又能表达底层代码级别的实现构件。同时基于该模型, 运用顺序通信进程, 提出了两级构件接口行为协议规约方法, 能够有效规约大粒度复杂软件构件的行为交互协议。

**关键词:** 基于构件的软件工程; 软件体系结构; 构件接口模型; 接口规约; 行为协议

## Architecture Oriented Component Interface Model and Its Formal Specification

REN Hongmin<sup>1</sup>, ZHANG Jingzhou<sup>2</sup>, QIAN Leqiu<sup>2</sup>

(1. Department of Computer Science and Technology, Shanghai Maritime University, Shanghai 200135;

2. Lab of Software Engineering, Department of Computer Science and Technology, Fudan University, Shanghai 200433)

**【Abstract】** Inspired by thoughts of software architecture, an architecture oriented component interface model (ACIM) is proposed. It can express not only high-level abstract components at architectural design, but also implementation components at low level design. Meanwhile, based on the component interface model and using CSP, two-level formal specification of component behavior protocols is proposed to describe the behavior protocols of large-scale components.

**【Key words】** Component based software engineering; Software architecture; Component interface model; Interface specification; Behavior protocol

基于构件的软件工程(Component based software engineering, CBSE)提高软件生产效率和质量, 日益成为软件开发的主流范型, 是软件开发实现工程化、自动化从而解决软件危机的现实可行途径<sup>[1]</sup>。

构件具有黑盒性质, 构件接口是构件开发人员和组装人员共同遵守的契约。但现今构件的工业标准, 如 CORBA、JavaBeans、COM 等, 它们的构件模型和接口定义语言(Interface definition language, IDL)仅仅规约构件接口的型构, 不能提供正确运用构件的足够信息, 阻碍了 CBSE 的发展和构件市场的形成。因此, 定义新的构件接口模型、扩充和增强构件接口规约成为 CBSE 研究的重要课题<sup>[2]</sup>。其中, 构件接口行为协议的形式化规约和分析成为构件接口扩充规约的研究热点<sup>[2]</sup>。

本文分析软件体系结构(Software Architecture, SA)领域和 CBSE 领域构件模型的特点, 提出了一种面向体系结构的构件接口模型(Architecture Oriented Component Interface Model, ACIM), 并基于该模型提出了两级构件接口行为协议规约方法。

本文是 SA 和 CBSE 结合的探索。它借鉴体系结构的研究思想和概念, 定义统一的构件模型, 贯通体系结构设计和 CBSE 构件组装实现。并运用层次结构控制构件行为协议规约的复杂度。

### 1 SA 和 CBSE 领域构件的特点

SA 和 CBSE 都有软件构件的概念, 但两个领域软件构件的概念和模型却各有特点, 并不完全一致。下面分析其各自的特点, 奠定这两个领域软件构件相互融合的基础。

SA 关注软件系统的全局组织结构, 是系统设计的蓝图。因此, SA 领域构件的第 1 个特点是抽象程度高、粒度大, 通常运用领域空间的术语进行描述。第 2 个特点是通常运用端口定义构件接口, 如 Wright<sup>[5]</sup>、插头插座式体系结构的构件模型<sup>[4]</sup>等。它的第 3 个特点是能够通过复杂的交互协议与其它构件交互。

CBSE 主张运用现有的软件构件组装生成能够执行的软件系统, 因此, CBSE 领域构件的第 1 个特点是面向实现, 能够执行, 第 2 个特点是遵守一定的包装(Packaging)准则, 支持异构环境的互操作。第 3 个特点是通用化但易于定制, 以支持复用。

### 2 面向体系结构的构件接口模型 ACIM

基于前面论述, 融合 SA 和 CBSE 的思想, 提出面向体系结构的构件接口模型 ACIM。

#### 2.1 ACIM 的概念模型

ACIM 的基本概念模型如下:

(1) 每个构件具有一个接口, 多个构件能够实现同一接口。通过接口理解和运用构件, 接口是构件提供自身信息并与外界交互的地点。

(2) 构件接口包含一个或者多个端口, 端口分为两类: 定制端口和交互端口。定制端口用于对构件自身功能进行定制

**基金项目:** 国家“863”计划基金资助项目(2001AA1100241)

**作者简介:** 任洪敏(1969—), 男, 博士、讲师, 主研方向: 软件复用和形式化方法; 张敬周, 博士生; 钱乐秋, 教授

**收稿日期:** 2004-10-11 **E-mail:** hmren@cie.shmtu.edu.cn

制,实现构件自身功能的完善,局部于定制的单个构件,不影响软件系统整体。交互端口用于构件与其它构件交互、通信,从而构件之间彼此协作,完成软件系统的整体功能。作个比喻,定制端口相当于计算机硬件板卡的跳线,而交互端口相当于计算机硬件板卡的插槽或插头。

(3) 定制端口包括数据参数、类型参数、函数参数或简单对象,通常由构件组装人员提供。定制端口不能包含向外提供服务元素。

(4) 交互端口包括共享的数据、服务提供操作、服务请求操作、向外发出消息、能够响应的外部消息、写入数据的管道、读出数据的管道,同时能够包括其它构件接口。

(5) 构件接口具有两级行为协议:端口一级行为协议和接口一级行为协议。ACIM 构件接口模型元素之间的关系如图 1。

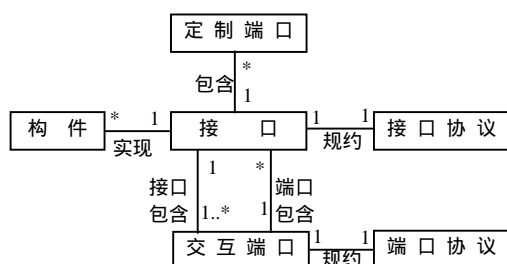


图 1 ACIM 构件接口模型元素之间的关系

## 2.2 ACIM 接口模型分析

ACIM 构件接口模型具有两个特点:一是定制端口和交互端口分类,二是交互端口和接口的嵌套定义。下面对其进行分析。

ACIM 进行端口分类,即定制端口和交互端口。虽然二者型构基本相同,但是它们的逻辑功能却不相同。定制端口用于构件定制,而交互端口用于构件交互和通信。并且定制端口经常在构件部署的时候设定,而交互端口能够与其它构件的端口进行动态交互。二者分离,符合多维关注分离的思想,易于构件理解、定制和复用。同时体现了 SA 的思想,即注重构件之间的逻辑交互关系。

SA 的研究把构件接口依据其逻辑交互功能划分为端口,是二级结构,而 ACIM 构件接口模型提出端口和接口相互嵌套定义,形成一个树型层次结构,是 SA 二级接口结构思想的深化。其遵循的思想是:复杂软件系统层次合成已经是普遍接受的原则,复杂的构件接口和交互协议同样应该能够层次合成和定义。由此带来的好处:控制接口规约复杂度;支持 SA 的求精与实现;提升构件组装的抽象级别等。

## 3 ACIM 构件接口模型的型构规约

下面运用 BNF 范式扩充表达方式,形式化地给出 ACIM 构件接口模型的型构规约。其中,具体低层的语法规约遵循 CORBA 接口定义语言,接口协议规约和端口协议规约见第 4 节。ACIM 型构规约如下:

```

component Interface Specification::=
    INTERFACE interface_name
    {
        [Interaction_port_section]+ //1 个或多个交互端口
        [Custom_port_section]* //0 个或多个定制端口
        [Interface_level_protocol] // 1 个构件接口级别的
    }
//交互协议定义

```

```

}
Custom_port_section::=
    CUSTOM_PORT portname
    {
        [Data_Parameter]* //0 个或多个数据参数定义
        [Type_Parameter]* //0 个或多个类型参数定义
        [Function_Signature_Declaration]* //0 个或多个函数参数
        [Object_Signature_Declaration]* //0 个或多个对象定义
        //上述诸项不能全部为空
    }
Interaction_port_section::=
    INTERACTION_PORT portname
    {
        [OWN Data_Definition]*
        //0 个或多个交互端口提供的、供交互对方共享的数据
        [SHARE Data_Definition]*
        //0 个或多个交互端口共享的、属于交互对方的数据
        [PROVIDE Operation_Signature_Declaration]*
        //0 个或多个提供服务操作定义
        [REQUEST Operation_signature_Declaration]*
        //0 个或多个请求服务操作定义
        [NOTIFY Message_Signature_Declaration]*
        //0 个或多个消息通知的定义
        [RESPONSE Message_Signature_Declaration]*
        //0 个或多个能够响应的消息定义
        [WRITE Pipe_Signature_Declaration]*
        //0 个或多个写入数据管道定义
        [READ Pipe_Signature_Declaration]*
        //0 个或多个读入数据管道定义
        [INTERFACE Interface_name Instance_name]*
        //0 个或多个接口实例定义
        [ DUAL INTERFACE Interface_name Instance_name]*
        //0 个或多个对偶接口实例
        //上述诸项不能全部为空。对偶接口定义见第 4 节。
        [Port_level_protocol] //1 个协议定义
    }

```

## 4 ACIM 构件模型行为协议规约

ACIM 构件接口模型从两个级别规约构件的行为协议:端口一级行为协议和接口一级行为协议。端口行为协议描述端口中元素的正确使用顺序和其间存在的依赖关系。接口行为协议描述接口的多个端口正确使用顺序和其间存在的依赖关系。在规约构件接口行为协议的时候,其包含的端口作为整体对待。在规约端口行为协议的时候,其包含的接口元素作为整体对待。两级行为协议规约结合接口端口嵌套定义,从而能够层次地规约构件行为协议。

CSP 提供丰富的行为描述机制,有完善的理论体系和成熟的支持工具,因此,ACIM 基于 CSP 描述构件接口行为协议。CSP 的概念和行为描述方法见文献[3]。

Wright 同样采用 CSP 描述构件的行为协议,但它自由定义构件的行为事件。而 ACIM 的特点在于基于构件接口型构的描述,确定构件的事件集合和描述构件的行为协议,从而协议规约变得简单和易于理解。下面给出两级构件行为协议规约中各种型构元素的事件定义,具体描述和运用方法见文献[3]。

**定义 1 (共享数据的事件)。**设共享数据名字为  $S_x$ ，则其事件、事件含义见表 1。

**表 1 共享数据名字为  $S_x$  的事件及事件含义**

事件 属性	OWN $S_x$	SHARE $S_x$
Read. $S_x$	OE。交互对方读数据 $S_x$ 。	IE。读交互对方的数据 $S_x$
Write. $S_x$	OE。交互对方写数据 $S_x$	IE。写交互对方的数据 $S_x$
Visite. $S_x$	简写，等价	简写等价 Write. $S_x$ Read. $P_x$

**定义 2 (操作调用的事件)。**设操作名字为  $O_x$ ，则其事件、事件含义见表 2。

**表 2 操作名字为  $O_x$  的事件及事件含义**

事件 属性	Provide $O_x$	REQUIRE $O_x$
Call. $O_x$	OE。交互对方调用操作 $O_x$ 。	IE。调用交互对方的操作 $O_x$
Return. $O_x$	IE。 $O_x$ 执行结束并返回	OE。 $O_x$ 执行结束并返回
$O_x$	简写，等价 Call. $O_x$ Return. $O_x$	简写，等价 Call. $O_x$ Return. $O_x$

**定义 3 (消息机制的事件)。**设消息名字为  $M_x$ ，则其事件、事件含义见表 3。

**表 3 消息名字为  $M_x$  的事件及事件含义**

事件 属性	SEND $M_x$	RESPONSE $M_x$
Send. $M_x$	IE。发送消息 $M_x$ 。	无意义
Response. $M_x$	无意义	IE。响应消息 $M_x$ 。

**定义 4 (管道机制的事件)。**设管道名字为  $P_x$ ，则其事件、事件含义见表 4。

**表 4 管道名字为  $P_x$  的事件及事件含义**

事件 属性	WRITE $P_x$	READ $P_x$
Begin. $P_x$	IE。开始管道传输数据。	IE。开始接受管道数据
End. $P_x$	IE。管道传输数据结束。	IE。管道读取数据结束
$P_x$	简写，等价 Begin. $P_x$ End. $P_x$	简写，等价 Begin. $P_x$ End. $P_x$

(上接第 62 页)

### 3 实例研究

例 2 设  $U = (2, 3, 4, 5, 4, 5, 6, 7, 8, 6, 6, 5, 4, 4, 5)$  为某股票连续 15 天(不包括星期六和星期日)的日收盘价。则  $|S(U, 4)| = 12$ ， $ssr(4) = \{\langle 2, 3, 4, 5 \rangle, \langle 4, 5, 6, 7 \rangle, \langle 5, 6, 7, 8 \rangle\}$ ， $ssf(4) = \{\langle 8, 6, 6, 5 \rangle, \langle 6, 6, 5, 4 \rangle, \langle 6, 5, 4, 4 \rangle\}$ ，极大的上升的子时间序列为  $\langle 4, 5, 6, 7, 8 \rangle$ ，极大的下降的子时间序列为  $\langle 8, 6, 6, 5, 4, 4 \rangle$ 。从  $U$  的极大的上升的子时间序列中得到规则：“该股票如果连续 5 个交易日的收盘价是升的，则接下去的那个交易日的收盘价是降的”；从  $U$  的极大的下降的子时间序列中得到规则：“该股票如果连续 6 个交易日的收盘价是降的，则接下去的那个交易日的收盘价是升的”。规则“该股票如果连续 4 个交易日的收盘价是升的，则接下去的那个交易日的收盘价是降”的支持度为  $3/12$ ，置信度为  $2/3$ ；规则“该股票如果连续 4 个交易日的收盘价是降的，则接下去的那个交易日的收盘价是升”的支持度为  $1/12$ ，置信度为  $1/3$ 。

### 4 结论

本文提出了新颖的时间序列模式和规则挖掘技术。本技术先把待挖掘的时间序列转换成子时间序列数据，然后利用

**定义 5 (端口执行和事件)。**端口执行是端口的一次逻辑交互，从而完成其定义的功能或子功能。设端口名字为  $P_x$ ，端口执行的开始是为完成该功能其包含的元素开始执行，用事件 Run. $P_x$  表示。端口执行结束是指该相关元素执行结束，用事件 End. $P_x$  表示。它的具体执行由该端口的端口协议决定，接口协议规约对其进行抽象。事件  $P_x = \text{Run.}P_x \quad \text{End.}P_x$ 。

**定义 6 (接口执行和事件)。**接口执行是接口的一次逻辑交互，从而完成其定义的功能或子功能。设接口名字为  $I_x$ ，接口执行的开始是为完成该功能其包含的端口开始执行，用事件 Run. $I_x$  表示。接口执行结束是相关端口执行结束，用 End. $I_x$  表示。接口中相关端口的执行具体由接口的行为协议规约，端口协议规约对其进行抽象。事件  $P_x = \text{Run.}P_x \quad \text{End.}P_x$ 。

### 5 结论

本文分析了 SA 领域和 CBSE 研究领域构件的特点，提出了面向体系结构的构件接口模型 ACIM，它既能表达体系结构设计的高层抽象构件，又能表达底层代码级别的实现构件，是 SA 和 CBSE 技术融合的一种探索和尝试。同时，基于 CSP，提出了 ACIM 的两级构件接口行为协议及其形式规约方法，能够简化复杂构件的行为协议规约和规约各种抽象级别构件的行为协议。

进一步的研究工作包括构件接口行为协议的分析、推导和自动调整机制的研究。

### 参考文献

- 1 杨芙清, 梅宏, 李克勤. 软件复用与软件构件技术. 电子学报, 1999, 27(2):68-75
- 2 Reussner R H. Enhanced Component Interfaces to Support Dynamic Adaption and Extension. In: 34<sup>th</sup> Hawaii International Conference on System Sciences, Maui, HI, IEEE, 2001-01
- 3 Hoare C A R. Communicating Sequential Processes. Prentice Hall, 1985
- 4 Luckham D, Vera J, Meldal S. Three Concepts of System Architecture. Technical Report, CSL-TR-95-674, Stanford University, 1995
- 5 Allen R. A Formal Approach to Software Architecture [D]. Carnegie Mellon Univ., CMU Technical Report CMU-CS-97-114, 1997-05

子时间序列所隐藏的知识，来指导对原时间序列的挖掘，从中提取模式或规则。给出了时间序列模式和规则的挖掘算法，并举例说明了本算法是有效和可行的。

### 参考文献

- 1 邵峰晶, 于忠清. 数据挖掘原理与算法. 北京: 中国水利水电出版社, 2003
- 2 Yoon J P, Lee J, Kim S. Trend Similarity and Prediction in Time-series Databases [A]. In: Proc. of SPIE on Data Mining and Knowledge Discovery: Theory, Tools, and Technology II. Washington: SPIE, 2000: 201-212
- 3 Agrawal R, Faloutsos C, Swami A. Efficient Similarity Search in Sequence Database. Springer Verlag, 1993: 69-84
- 4 Das G, Gunopulous D, Mannila H. Finding Similar Time Series. In: Proc. of 1<sup>st</sup> European Symposium on Principles of Data Mining and Knowledge Discovery(PKDD'97), Komorowski J, Zytkow J(Eds.), 1997
- 5 Box G, Jenkins G. Time Series Analysis: Forecasting and Control. San Francisco, CA: Holden-Day, 1970