计算机科学2003Vol.30№.2

软件体系结构描述研究与进展

孙昌爱　金茂忠
（北京航空航天大学计算机科学与工程系　北京100083）

Overview on Software Architecture Description

SUN Chang-Ai　JIN Mao-Zhong
(Dep. of Comp. Sci. & Eng., Beijing Uni. of Astro. & Aero., Beijing 100083)
Email:sca@safepro.buaa.edu.cn and jmz@buaa.edu.cn

Abstract　Recently, Software Architecture has become one focus in software Engineering community, and a key issue to successful large-scale software development. Software Architecture Description forms the base for architectural construction, evolution, verification, analysis, maintenance and architecture-driven software development. The concept related with Software Architecture Description is introduced, and Architecture Description Language and Typical Architecture Description Method are discussed in the paper. At the same time, the industrial standards for Software Architecture Description, such as IEEE P1471 and Rational RAS-ADS, is also introduced. Finally, the conclusion and envision for Software Architecture Description are present in the paper.

Keywords　Software architecture, Software architecture description, Architecture description language, Software architecture documentation, Architecture description standard, Architecture reuse

软件体系结构[1,2]是当前软件工程领域的一个研究热点，是大型软件开发中必须解决的核心技术。无数的软件工程实践也证明了：一个成功的软件系统往往都有一个好的软件体系结构。由于软件体系结构描述是体系结构构造、演化、验证、分析、维护和基于体系结构的软件开发的基础，因此体系结构描述方法是该领域中的一个主要研究内容（通常，体系结构是指系统体系结构描述，包括了软件单元、硬件单元以及两者的映射关系。本文在不加区分的情况下，体系结构是指软件体系结构）。

1　软件体系结构描述的基本问题

1.1　几个重要概念

软件体系结构　通过对诸多软件体系结构定义的研究和归纳[3]，不难发现各种体系结构定义基本上都涵盖如下实体：

·构件，包括计算构件和数据构件；

·连接件，实现构件之间的交互、通讯或协调关系；

·限制，包括构件通讯协议、实时性、同步等；

·配置，反映系统的总体结构，构件和连接件构成的拓扑结构；

·设计原则与指导方针，即体系结构设计或描述的知识，包括体系结构风格、模式等高级概念。

因此，可以将软件体系结构定义为"从一个较高抽象层次来考虑组成系统的构件、构件之间的交互，以及由构件与构件交互形成的拓扑结构的关系。这些要素应该满足一定的限制，遵循一定的设计规则，能够在一定的环境下进行演化。软件体系结构应能反映系统开发中具有重要影响的设计决策，便于各种人员的交流，反映多种关注，照此开发的系统能完成系统既定的功能和性能需求。"[3]

体系结构描述　软件体系结构本身是一种概念，描述体系结构的过程称为体系结构构造，以体系结构描述语言刻画体系结构的结果称为体系结构表示[4]。

体系结构模型　从抽象角度对系统结构的完整的描述[5]。

体系结构视图　从一个特定的角度或者着眼点对系统进行一种简化的描述（或者抽象），描述只需覆盖关注的方面，而忽略与此无关的实体[5]。视点（Viewpoint）是指有选择的体系结构构造子或者结构规则集合的一种抽象形式，通常表达系统的某个方面[4]。一般来说，体系结构视点是体系结构视图的模板或者模式，指明了体系结构视图的目的和面向的角色，并提供了构造视图的技术与方法。

通常，体系结构描述是由反映系统各种人员及其关注的视图组成。一个体系结构要从多层次、多角度、多阶段进行描述，每个视图只能属于一种视点。视点与视图的关系就如同类与对象的关系；而视图与体系结构描述的关系如同章节与书的关系。根据本文给出的软件体系结构概念，不难得出体系结构描述的概念框架，如图1所示。

1.2　软件体系结构描述研究的必要性

在以往实践中，尽管软件体系结构客观存在，但往往被忽视。由于软件体系结构对软件生命周期具有重要的影响，并随着软件开发费用的不断提高、软件规模及复杂性不断增加，软件体系结构成为系统能否开发成功的关键因素之一。但是我们知道，软件体系结构仅仅是人们对软件高层抽象结构的概念，软件体系结构描述则是希望将这种概念上的东西表述为可见的文档。体系结构描述具有如下作用：

1）支持系统不同人员之间的通讯；

2）能够表达系统及其演化；

孙昌爱　博士生，主要研究领域为软件体系结构与构件技术，面向对象技术，软件测试技术、软件工程环境。金茂忠　教授，博士生导师，主要研究领域为软件工程环境，软件测试。

3)体系结构评估、分析与比较的基础；

4)便于系统开发活动的计划、管理与执行；

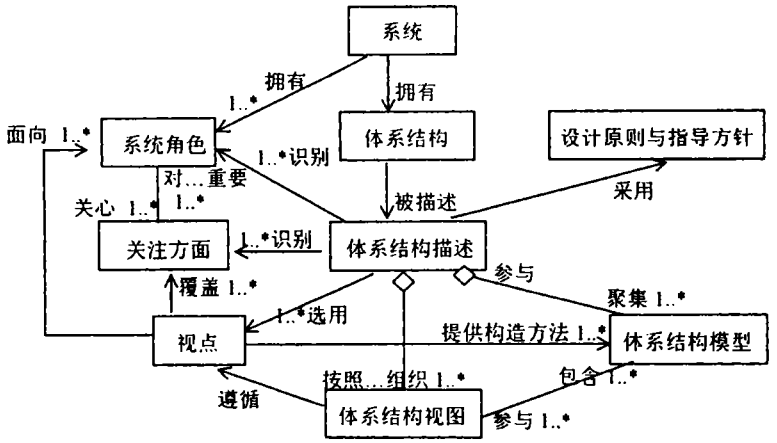


图1 软件体系结构描述的概念框架示意图

5)表达了系统的持久化特征和所支持的基本原则，有利于系统的进一步变化；

6)便于体系结构的实现与描述之间的一致性检查；

7)记录了体系结构构造的决策知识；

8)系统开发的一种重要文档，中间产品的一部分。

1.3 体系结构描述的范畴

一个软件体系结构应该描述哪些内容？系统的结构属性应该是软件体系结构构造和描述的核心内容[6]。但是，在具体描述内容上，没有统一的结论。

Booch 认为软件体系结构描述应该着重识别、选择与确认那些对体系结构有重要影响的元素[5]：1)主要的业务类；2)重要的机制；3)处理与过程；4)分层与子系统；5)接口。

Mary Shaw 等认为，本质上软件体系结构是对系统整体结构设计的刻画，因此软件体系结构应该着重设计与描述如下内容[2]：1)总体组织与全局控制结构；2)构件间通讯、同步和数据访问的协议；3)设计元素间的功能分配；4)物理分布；5)设计元素复合；6)伸缩性和性能；7)设计选择等内容。

根据定义“SA＝{elements，form，rational}”[1]和定义“SA＝{component，idioms / styles，common patterns of interaction}”[7]，那么在其概念框架下的软件体系结构描述内容应包含体系结构风格、设计规则等内容。

按照本文给出的软件体系结构定义及其体系结构描述的概念框架，软件体系结构应该描述：1)以软件结构信息为中心的元素、元素之间的关系，以及指导这些元素与元素关系的指导原则；2)对结构信息有重要影响的系统相关属性，包括功能分配，物理分布，软硬件间的映射。具体说来，软件体系结构应该描述：1)总体结构与控制方式；2)构件；3)构件之间的交互；4)构件与构件交互之间的功能分配；5)构件及构件的复合机制，抽象机制等；6)构件与构件交互的物理分布。

鉴于体系结构描述的上述重要性，并由于对体系结构描述的范畴的不同理解，因而也出现了多种体系结构描述语言和不同的体系结构描述方法。

2 软件体系结构描述语言 ADL

2.1 ADL 的发展过程

ADL 不同于程度设计语言，它是一种刻画体系结构层的元素、元素之间的关系规格说明语言。体系结构描述语言的发展存在如下几个阶段：

1)矩形框和有向线段的组合：这是一种最原始也是最常见的体系结构描述方式，矩形框表示系统的构件或者模块，而有向线段表示构件或者模块之间的关系。由于缺乏必要语义约束，因此不能反映体系结构中的复杂关系；

2)模块互连语言：结构化程序设计语言的延伸，通过导入/导出实现模块之间的交互关系，如 Ada/Pascal 语言采用 use 实现包/过程(函数)模块的复用或交互；

3)基于软构件的系统描述语言：扩展反映系统构成的描述语言，使其支持体系结构描述，如一种多变配置语言 PCL (Proteus Configuration Language)[8]就可以用来在一个较高的抽象层次上对系统的体系结构建模，Darwin[9]最初用作设计和构造复杂分布式系统的配置说明语言，但因具有动态的特性，可用来描述动态体系结构；

4)ADL：已有几十种常见的 ADL[10]。

1)、2)、3)三种描述方式存在较多的二义性，而且用法不统一，缺乏对精确的体系结构概念和判断准则支持，因而无法进一步进行体系结构分析、合成和细化。

2.2 ADL 的设计、分类和比较框架

ADL 应该满足体系结构建模的需要，因此 Tracz 认为 ADL 应包括4“C”，即构件、连接件、配置和约束[11]。而 Mary Shaw 等认为，ADL 应具备程序设计语言的严谨、精确之外，还应具有如下特点：构造、抽象、重用、组合、异构、分析推理能力[2]。

通过对众多常见 ADL 详细的分析和归纳，Medvidovic 给出了 ADL 的概念与框架[10]，并认为一个体系结构应该至少包括如下四个元素：应该具备明确地对构件、连接件、配置建模的能力，且提供适当的工具支持。其中：

1)构件＝{接口，类型，语义，约束，演化，非功能特性}；

2)连接件＝{接口，类型，语义，约束，演化，非功能特性}；

3)配置＝{体系结构配置描述的质量＝{可理解性，可组装性，细化和可追踪性，异构性}，所描述系统的质量＝{异构性，可扩展性，演化性和动态特性}，所描述系统的特性＝{动态特性，约束，非功能特性}}；

4)支持工具＝{主动的规格说明，多视图，分析，细化，实现产生，动态特性}。

2.3 ADL 的不足之处

尽管研究并提出了若干的 ADL，典型的有 C2、Darwin、UniCon、LILEAnna、P++、LEAP、Wright、uRapid 等[2,10]。但

大多数 ADL 只能描述某一个特定的应用领域的软件体系结构，还不存在一个通用的 ADL。为此，出现了不同 ADL 所说明的体系结构规格说明之间的转换问题，Mary Shaw 等设计了一种支持不同 ADL 之间互换的体系结构语言 ACME[12]。

分析现有的 ADL 可以发现：形式化的 ADL 采用严格的形式化规格说明，可以支持严格的验证和推理，但体系结构描述不够直观，而且对体系结构设计师要求较高；可视化的 ADL 缺少严格的语义约束，验证困难，支持进一步的分析困难。

3 软件体系结构描述方法

3.1 体系结构描述的基本原则

尽管体系结构描述语言各不相同，描述方法也有所不同。结合 IEEE P1471标准，我们认为，软件体系结构描述应该满足或支持如下目标：

1)能够从体系结构上表达系统及其演化；

2)支持系统人员之间的交流及其各自关注；

3)能以一致的方式评价和比较软件体系结构；

4)验证系统的实现是否和体系结构描述一致；

5)记录与表达软件体系结构的一些成熟的知识(如体系结构风格)；

6)能够区分体系结构和非体系结构方面的因素。

3.2 几种软件体系结构描述思路

ADL 为体系结构描述提供了工具，但是如何描述以及描述哪些方面则是体系结构描述方法的范畴。现有的体系结构描述思路可以分为：形式化描述方法和可视化建模方法。

3.2.1 *形式化体系结构描述方法* 体系结构描述应该足够严格严谨，支持体系结构推理和验证，以及性能的量化分析。提出了如下几种比较典型的形式化体系结构描述：

1)基于图文法的体系结构描述[15]：将体系结构描述看成图的构造，图中的结点表示构件，边表示构件之间的交互，构件和构件交互的约束则表示成结点和边的属性，图的拓扑结构表示体系结构配置；体系结构生成和演化的过程则表现为图的置换；

2)化学抽象反映机制 CHAM[16]：采用类似于化学反应的原理来描述体系结构的动态特性，将构件看成化学反应中的分子，构件交互/通讯/协调看成反应规则，因此，体系结构变换的过程类似于具有初始态的溶剂(构件初始值集合)，经过一系列的反映规则的变换(构件交互)，生成目标溶剂状态(目标的体系结构)；

3)多值代数的体系结构描述[17]：从多值代数变换的角度描述体系结构变换，即$(a_1,a_2,\cdots,a_n)-->(b_1,b_2,\cdots,b_m)$，其中变换式左边表示变换前的体系结构属性值向量，而右边则表示变化后的体系结构属性值向量；

4)基于公理系统的体系结构描述[18]：将体系结构看成是一个理论，可以采用一阶谓词逻辑结合类型理论描述构件、连接件，而采用不变式断言表达体系结构配置，软件需求对体系结构约束则表现为理论中的公理。

3.2.2 *可视化体系结构描述* 体系结构应该便于各种人员的交流，是系统的开发蓝图，因此体系结构应采用可视化的描述方法，并且通常是多视图描述方法。绝大多数的 ADL 都支持可视化体系结构描述，例如 C2，Darwin。

可视化体系结构描述方法的一个典型代表是 Kruchten 提出的“4+1”的软件体系结构描述模型[19]。该模型由逻辑视图、开发视图、进程视图、物理视图和用例视图构成，分别从功能、静态结构、动态行为、部署四个角度描述体系结构，并且用场景将上述四个视图关联起来。

可视化体系结构描述可能导致不一致的体系结构文档，解决的方法是：在现行的可视化体系结构语言和标记的基础上，引入一些语义限制，如 C2采用了绑定的方法将可视化图法表示与形式化的语义(如体系结构修改语言 AML)结合起来[13]；Darwin 在可视化标记的基础上引入了支持动态修改的脚本语言；在可视化建模语言 UML 中引入 OCL[14]则可以有效地改进语言的规范性，同时不失可视化的特点。

3.2.3 *扩充 UML 描述软件体系结构* 形式化规格说明语言、面向对象的建模表示、高层的设计表示和模块互连语言与 ADL 很相似。但是，根据 ADL 的定义和比较框架[12]，它们都不属于 ADL。UML 是一种可视化的面向对象的建模语言，而且与 ADL 有着相似的软件建模哲学，但二者有着不同的假设：UML 倾向的用途并不是用来体系结构建模；UML 中的一些假设在体系结构描述语言中则不存在。尽管如此，UML 仍具有较强的体系结构建模特性，如表1。因此，在当前的体系结构描述实践中，若干研究者都提出了用 UML 描述体系结构的思路。

表1 UML 的体系结构建模特性分析

体系结构元素	相应的 UML 元素
构件	类、构件、节点、用例、包、子系统
连接件	关系，包括泛化、关联、依赖
接口	接口
约束	规则，包括 OCL
配置	构件图、包图、配置图
其它属性	扩展机制

3.2.3.1 用 UML 描述体系结构的两种思路

1)直接 的 UML 体系结构描述：根据某种体系结构描述模型，如4+1模型，然后以 UML 中类似的模型去表示与实现。例如，用 UML 进行分布式企业系统体系结构建模就属于此种类型[20]；

2)间接的 UML 体系结构描述：根据一种 ADL 的元素和描述思路，采用 UML 模型元素来表示这些元素和描述，这种体系结构的 UML 描述是间接的，要考虑 ADL 向 UML 的转换。例如，采用 UML 来模仿 Wright 语言的体系结构描述[21]。

3.2.3.2 三种 UML 实现体系结构建模方法及其比较

1)无扩展方法(直接的 UML)：按照原来的方式使用 UML，并且直接比较 UML 和某个 ADL，说明体系结构与设计之间的关系。因而在体系结构建模和软件建模之间存在一种映射关系，在 UML 描述的体系结构到以 UML 描述的设计之间实现转换。该方法描述的体系结构是可理解的，可以有多个标准的工具进行操作，但是可能违背体系结构约束；

2)内嵌扩展方法(约束的 UML)：应用 UML 对元类所支持的扩展机制，选择在语义上与 ADL 构造子相近的元类，定义一个构造型，并将其应用到元类实例上，但类的语义应该遵循 ADL 的约束，允许一致性检查。该方法保证了体系结构的约束，但需要完全的风格规格说明，需要 OCL 兼容的工具；

3)定制扩展方法(扩充的 UML)：扩展 UML 直接支持体系结构问题，即直接增加体系结构描述的设计元素。该方法对体系结构提供原本的支持，但是需要后向兼容工具支持，会导致 UML 版本的不兼容。

3.3 体系结构描述方法比较

基于不同的目标，各种体系结构描述方法具有不同的特点：

1)形式化体系结构描述方法：关注体系结构模型的解析评估和特定的解决方案，因而采用单个模型和严格的建模符号；体系结构描述精确，支持强有力的分析；

2)可视化体系结构描述方法：关注广泛范围内的开发问题，强调的是实践可行性而非精确性，因而采用多视图的可视化的体系结构描述模型，具有直观、便于交流的特点；

3)扩展的 UML 体系结构描述方法：UML 已经成为工业界的建模语言标准，有强大的工具支持，有利于体系结构在软件开发中的广泛应用。

3.4 软件体系结构描述存在的问题

ADL 和体系结构描述方法研究非常丰富，但是体系结构描述仍然存在诸多的不足之处：首先，体系结构描述尚缺乏一个统一框架，导致在不同的体系结构描述语言及其体系结构描述方法得到的体系结构规格说明之间无法实现有效的复用。Medvidovi 提出的体系结构描述语言的概念框架和分类标准，以及工业界建议的体系结构描述框架标准有利于该问题的解决。其次，体系结构描述的形式化和可视化的对立，导致精确性和可用性形成一对矛盾。最后，在多视图的体系结构描述中，体系结构描述的完备视图集合也无法达成共识。在大多数体系结构描述方法中，都基于体系结构视点来描述软件体系结构。但是对视点的选择缺乏统一的标准。例如，常见的体系结构视点包括：1)结构视点；2)行为视点；3)物理互连视点；而在可复用资产的规格说明中的体系结构描述则采用了四种视点：1)需求视点；2)设计视点；3)实现视点；4)测试视点。

4 软件体系结构描述框架标准 IEEE P1471

鉴于体系结构描述的概念与实践的不统一，IEEE 于1995年8月成立了体系结构工作组，综合体系结构描述研究成果，并参考业界的体系结构描述的实践，负责起草了体系结构描述框架标准即 IEEE P1471[4]，并于2000年9月21日通过 IEEE-SA 标准委员会评审。

IEEE P1471适用于软件密集的系统，其目标在于：便于体系结构的表达与交流，并通过体系结构要素及其实践标准化，奠定质量与成本的基础。本标准详细介绍了一套体系结构描述的概念框架，并给出建立框架的思路。同时，该标准还讨论了体系结构描述实践，在应用体系结构描述的推荐标准时，应该遵循如下几个具体的要求：

1)体系结构的存档要求；2)能识别人员及其关注；3)体系结构视点的选择(视点的具体规格说明)；4)体系结构视点；5)体系结构视点之间的一致性；6)体系结构原理。

IEEE P1471仅仅提供了体系结构描述的概念框架、其体系结构描述实践应该遵循的规范，但如何描述以及具体的描述技术等方面缺乏更进一步的指导。

5 可复用的软件体系结构描述

在 IEEE P1471推荐的体系结构描述的概念框架基础上，Rational 起草了可复用的软件资产规格说明，专门讨论了体系结构描述的规格说明[22]，提出了一套易于复用的体系结构描述规范。该建议草案已经提交 OMG，可望成为体系结构描述的行业规范。

可复用的体系结构描述框架建议，基于 RUP、采用 UML 模型描述软件的体系结构，认为体系结构描述的关键是定义视点、视图以及建模元素之间的映射关系。可以从四个视点出发描述体系结构，即需求视点、设计视点、实现视点和测试视点。并在此基础上提出了7个体系结构视图，即用例视图、域视图、非功能需求视图、逻辑视图、实现视图、过程视图和部署视图。然后，从系统建模的角度考虑多个视图之间的影射关系，并建议了这些视图的表示和视图之间的影射关系的表示。

本建议标准采纳了 IEEE P1471中提出的体系结构描述概念模型，覆盖了"4+1"体系结构模型中的4个视图，并在原则上讨论了视图以及视图之间的映射的表示问题。与 IEEE P1471相比，该建议标准的体系结构描述方案涉及面比较窄，所注重的层次比较低，因而更具体。由于将体系结构的描述限于 UML 和 RUP，具有一定的局限性，但该建议标准结合了业界已经广泛采用的建模语言和开发过程，因而易于推广，可以有效实现在跨组织之间复用体系结构描述结果。

小结 体系结构描述的结果表现为体系结构规格说明，是软件开发过程中的一个极其重要的文档。体系结构描述的核心内容：什么是体系结构描述、为何需要体系结构描述、体系结构描述的对象是什么、体系结构描述的工具有哪些(即 ADL)、ADL 的比较及存在的问题、体系结构描述的基本原则、体系结构描述的典型方法、描述方法的分析比较等一系列问题。而在实践中，体系结构描述的概念框架标准 IEEE P1471，可以推动体系结构描述的概念和实践统一；采纳 Rational 可复用资产规格说明中的体系结构描述框架建议标准(RAS-ADS)，则可以进一步实现体系结构描述的复用。

根据当前的体系结构描述的现状，我们认为，几个可行的体系结构描述方向为：1)基于统一的体系结构描述概念框架，开发统一的体系结构描述语言及其支持环境；2)实用的体系结构描述方法；3)ADL 描述的体系结构规格说明向 UML 模型的转换；4)在 UML 支持环境中增加对体系结构建模的支持。

参考文献

1 Perry D E, Wolf A L. Foundations for the study of software architecture. ACM SIGSOFT Software Engineering Notes, 1992, 17(4): 40～52

2 Shaw M, Garlan D. Software Architecture. Prentice Hall, 1997

3 孙昌爱，金茂忠，刘超. 软件体系结构研究综述. 软件学报(已提交)，2001

4 IEEE's Recommended Practice for Architectural Description. IEEE P1471-2000

5 Booch G. Software Architecture and the UML. from http://www.Rational.com/publications, 1993.3

6 Clements P C, Northrop L M. Software Architecture: An Executive Overview. from http://www.sei.cmu.edu/publications/Software Architecture: An Executive Overview.pdf

7 Vestal S. A Cursory Overview and Comparison of Four Architecture Description Languages, Honeywell Technology Center technical report, February 1993. http://www-ast.tds-gn.lmco.com/arch/four-adl.ps

8 Sommerville I, Dean G. PCL: a language for modeling evolving system architectures. Software Engineering Journal, 1996, 11(2): 111～121

9 magee J, Kramer J. Dynamic Structure in Software Architectures. In: Proc. ACM SIGSOFT'96: Fourth Symp. Foundations of Software Eng. (FSE4), Oct. 1996. 13～14

(下转第167页)

所有参数与前面相同，样本数目比例依然为80:20。

采用上面介绍的新算法，并在计算过程中使

$$\frac{C_1}{C_2}=\frac{1}{4} \tag{13}$$

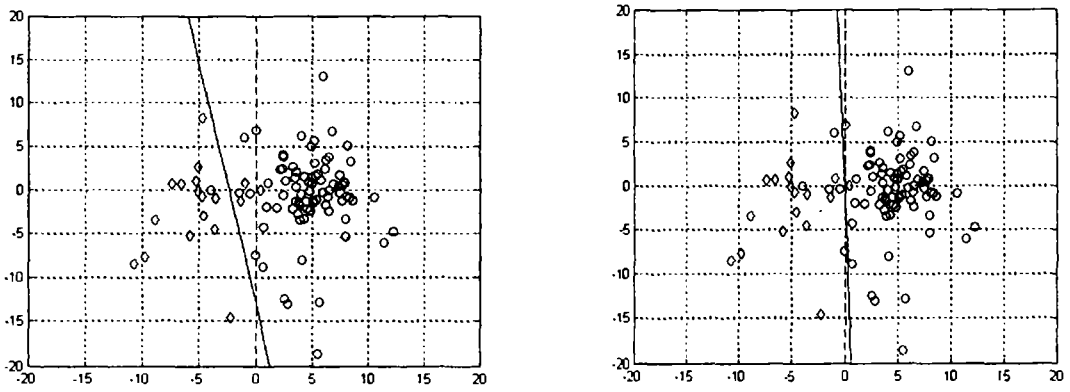
图3为运算结果，实线为实际分类线，点划线为期望的分类线。将图3与图2(b)比较，不难发现，新分类线更为接近于期望的分类线。

算例2 两维两类样本，分别服从指数分布

$$p(X)=\frac{1}{2}\lambda\exp[-\lambda\|X-\mu\|] \tag{13}$$

式中 $\lambda=4$，对于类别一，$\mu_1=\begin{bmatrix}5\\0\end{bmatrix}$，样本数目为80；对于类别二，$\mu_1=\begin{bmatrix}-5\\0\end{bmatrix}$，样本数目为20。

图4(a)，(b)分别画出了在一般 SVM 模型和 SVM 新模型下的分类线，从图中可以看出，新模型克服了样本数目差异所造成的影响。



(a) 一般 SVM 模型　　(b) SVM 新模型

图4　指数分布下，样本数目比例不同时的 SVM 模型比较

结论　本文简要地分析了 SVM 的分类原理，指出在样本数目相差较悬殊时，SVM 不能获得良好的分类能力。为此，本文提出了对不同类的错分样本进行不同惩罚的方法，两个算例的运算结果也充分说明，这种修正是有效的。

为简化描述，本文采用最为简单的向量内积作为核函数，设计上，整个算法对所有的核函数都是适用的。

参 考 文 献

1 Vapnik V. The Nature of Statistical Learning Theory. New York:Springer-Verlag,1995

2 Osuna E,Freund R,Girosi F. Support Vector Machine:Training and Applications. AI Memo 1602. MIT Artificial Intelligence Laborabory,1997

3 Burges C J C. A Tutorial on Support Vector Machines for Pattern Recognition. Data Mining and Knowledge Discovery,1998,2(2):121～167

4 Campbell C. Kernel Methods:A survey of Current Techniques. http://citeseer.nj.nec.com/compbell00kernel.html

5 Schölkopf B,Smola A,Müller K R. Nonlinear Component Analysis as a Kernel Eigenvalue Problem. Neural Computation,1998,10:1299～1319

6 Ruiz A,López-de-Teruel P E. Nonlinear Kernel-Based Statistical Pattern Analysis. IEEE Trans. on Neural Networks,2001,12(1):16～32

7 张学工. 关于统计学习理论与支持向量机. 自动化学报,2000,26(1):32～42

8 肖健华,樊可清,吴今培,等. 应用于故障诊断的 SVM 理论研究. 振动、测试与诊断,2001,21(4):258～262

(上接第139页)

10 Medvidovic N, Taylor R N. A Classification and Comparison Framework for Software Architecture Description Languages. IEEE Transaction on Software Engineering,2000,26(1):70～93

11 Tracz W. LILEANNA:A Parameterized Programming Language. In:Proc. Second Int'l Workshop Software Reuse,Mar. 1993. 66～78

12 Garlan D,Monroe R,Wile D. ACME:An Architecture Description Interchange Language. In:Proc. CASCon'97,Nov. 1997

13 Medvidovic N,Rosenblum D S,Taylor R N. A Language and Environment for Architecture-Based Software Development and Evolution. In:Proc. 21st Int'l Conf. Software Eng. (ICSE'99),May,1999. 44～53

14 OMG AD/97. 08. 05,1997,United Modeling Language Specification,version 1.1,Object Management Group

15 Metayer D L. Describing software architecture styles using graph grammars. IEEE Transactions on Software Engineering,1998,24(7):521～533

16 Inveradi P,Wolf A L. Daniel Yankelevich,Behavioral Type Checking of architectural Component Based on Assumptions. From:http://www.sei.cmu.edu/publications/cu-cs-861-98.ps

17 Yuan Chun,Chen Yiyun. Software Architecture Evolution by Multiset Transformation. In:Proc. of Intl. conf. on Software:Theory and Practice,Aug. 2000. 236～243

18 云晓春,方滨兴. 基于构件设计的正确性验证. 小型微型计算机系统,1999,20(5):330～334

19 Kruchten P B. the 4+1 View Model of Architecture. IEEE Software,1994. 42～50

20 饶若楠,尤晋元. 基于统一建模语言的分布式企业信息系统软件体系结构模型. 上海交通大学学报,2001,34(7):979～982

21 邓勇,丁峰,沈均毅. 基于 UML 的软件体系结构建模方法研究. 小型微型计算机系统,2001,22(10):1206～1209

22 Rational,Reusable Asset Specification—— Architectural Description Standard:[Technique Report]. from *http://www.Rational.com/publications*