

# Web 应用开发的两层体系结构建模

Two-Level Architecture Model for Web Applications Development

杨卫东 施伯乐

(复旦大学计算机科学系 上海200433)

**Abstract** With the rapid development of Internet, the Web has evolved into a main platform for delivering all kinds of applications. The complexity of Web applications requires more systematic method and model to support it. This paper describes development process of Web applications, presents a two-level architecture model of Web applications and its modeling language(object-oriented modeling language--OOWML), which includes hypermedia model and data model. With the Object-Oriented concepts, this method uses object as an uniform modeling entity, provides high-level description of a Web application, supports modeling Web service entities at arbitrary levels of granularity, abstraction and reuse. OOWML is a textual language based on XML and independent of specific platform, supports design, implementation and maintenance for Web applications.

**Keywords** Objects, Web applications, Object-Oriented Web modeling language, Data model, Hypermedia model, Software reuse

随着 Internet 的迅猛发展, Web 已经成为各种应用的主要平台。在 Web 应用的开始阶段, 只是一些简单的网站, 将 Web 作为信息中介和信息发布的平台, 这时候所提供的工具只是集中于网页的创建和编辑, 如 HotMetal 编辑器和微软的 FrontPage, 它们都限于被动的超文本表示以及一致的跨网页的布局和部分的导航模型。随着网站的规模和复杂度日益增加, 它所承担的任务已不仅是简单的基于 Web 的数据发布, 也包括了信息系统的构建和复杂的应用逻辑, 我们将这一类应用统称为 Web 应用。我们可以看到, 对于 Web 应用, 早期的网站编辑工具难以支持系统的开发、管理和维护。近年来的一些研究, 借助于软件工程领域中的开发方法和模型, 以支持 Web 应用的系统化、结构化的开发, 这类方法称为模型驱动的开发方法。近年来, 模型驱动的开发方法成为 Web 应用开发和建模的研究重点。其中较突出的有 RMM 方法<sup>[4]</sup>、Areneus 方法<sup>[11]</sup>、OOHDM<sup>[5]</sup>方法和 WebComposition 方法<sup>[6]</sup>等, 其中 RMM 方法和 Areneus 方法以实体联系图描述概念模型的主要手段, 建立在结构化开发方法的基础之上。RMM 较好地支持了 Web 应用开发的整个生命周期, Areneus 方法在数据库和 Web 的集成和网站的重建方面较为突出, 同时考虑了不同网站的集成问题。但是, 缺乏从企业应用的角度考虑模型和开发方法, 如考虑软件的体系结构和软重用等。由于面向对象技术已在工业界广泛采用, 包括后端的 Web 开发技术和数据库管理系统; 再者, 面向对象更能反映企业应用, 因此, 关于面向对象的 Web 方法和模型的研究更受到人们的关注, 其中, OOHDM 方法具有代表性。OOHDM 方法支持 Web 应用开发的整个生命周期, 强调了 Web 应用的超媒体特性(如导航结构的建模和设计)。Web-Composition 建立在面向对象的基础之上, 以构件做为 Web 应用建模的基本实体, 但是缺乏对概念模型的支持。

本文提出了 Web 应用开发的两层体系结构模型及其面向对象的建模语言(OOWML)。Web 开发模型分为两层: 超媒体模型和数据模型, 超媒体模型描述了 Web 服务的内容和表示, 并显式地将数据的内容和表示分开, 数据模型描述了应用领域的信息结构, 并包含类似于 OQL 的表达式以利于数

据内容的导出。该模型具有以下特点:

- 它以面向对象的概念支持 Web 应用的建模, 涵盖了 Web 应用开发的整个生命周期, 从设计到实施提供了操纵 Web 资源一致的接口;
  - 集成了静态的信息网页、多媒体数据类型和交互的 Script, 更好地支持动态 Web 应用的开发;
  - 数据模型和超媒体模型都基于 XML 的标记描述语言, 与特定的平台无关, 利于 Web 开发模型的互操作;
  - 在更为抽象的层次上支持 Web 应用的建模, 支持 Web 对象的重用、布局和信息内容的分离, 简化了复杂 Web 应用的管理和维护;
  - 面向对象的 Web 模型和建模语言 OOWML 的编译器将抽象的 Web 对象模型自动映射为基于文件的 Web 资源; 支持 Web 应用快速原型的开发和迭代的开发过程。
- 本文的一个直接需求是, 复旦大学计算机系数据库组承担的国家自然科学基金重点项目“数字图书馆关键技术研究”(69933010)中的相关网站建模、数据发布和站点维护, 并受到该项目的资助。本文的组织如下: 第一节简要介绍了 Web 应用开发过程; 第二节简要介绍了面向对象的 Web 开发模型; 第三节介绍了 OOWML, 包括两部分: 数据模型的描述和超媒体模型的描述; 第四节简要介绍了 OOWML 的实施; 第五节总结了本文的工作, 提出了准备进一步研究的内容。

## 一、Web 应用的开发过程

Web 应用越来越复杂, 需要类似于传统应用的软件开发方法的支持软件开发的整个生命周期。关于 Web 应用开发生命周期, 尽管还没有统一的开发过程和模型, 但都应该包括以下活动<sup>[5-8]</sup>:

- 需求分析。需求分析主要确定系统的主要任务和可行性评估。
- 概念设计。类似于信息系统的设计, 着重于建模应用领域中的对象和它们之间的关系。
- 内容设计及导航设计。内容表示了对用户显示的数据内容, 导航设计着重于考虑所表示内容之间的超文本链接。

·表示设计,表示了具体网页的布局和显示信息。

·评估和测试。初步的原型系统需要提交给用户,收集用户的反馈信息,以修正相应的设计。另外,需求的改变和错误的修订也需要对结构、表示、导航等的修改。

在 Web 应用的背景下,由于特定于 HTML 的用户界面,包含了复杂的导航和布局信息,需要不断地进行用户评估、收集用户信息反馈,因此,在 Web 应用的开发过程中,支持原型的快速开发非常重要。另外,在开发过程的每一个活动中,都有可能返回到前面的活动进行设计的修改,因此,Web 应用的开发更适合于迭代式的开发过程。

## 二、面向对象的 Web 开发模型

与传统的应用系统开发相比,Web 应用开发缺乏系统的、结构化的方法和模型支持。我们建立 Web 应用开发模型强调了以下几点:

(1)Web 应用具有特定的体系结构,通常是两层或多层体系结构。与传统的 Client-Server 结构相比,它大都是基于浏览器的“瘦”客户,更多的是负责信息的显示和布局,应用逻辑和数据都放在服务器端;同时,Web 应用的开发通常具有明确的技术分工,如专门的美工负责信息的布局和显示,专门的技术人员负责网页的组装和导航,信息专家负责应用领域的信息结构和组织。因此,我们将 Web 开发模型组织成不同的层次或纬度。

(2)Web 应用是以基于文件的 Web 资源来交付,缺乏抽象的概念,使得开发和维护比较困难。因此,我们用面向对象的概念抽象地描述 Web 应用,并为 Web 应用开发生命周期提供一致的模型。

(3)Web 应用的开发需要支持重用的机制,例如,我们经常遇到的一种情况是,在 Web 实施中,要经常复制和粘贴同样的资源,对它们的修改很容易导致不一致性。我们通过 Web 对象之间的继承和组装关系,有效地支持重用。

根据 Web 应用的特点,我们采用了类似于面向对象的设计模板 MVC(Model/View/Controller)<sup>[7]</sup>,将 Web 模型分为不同的层次:

·数据模型:描述了应用领域的信息结构和组织,它由类以及类之间的关联表示,通常由数据库系统来管理。该模型通常首先建立,或者通过逆向工具从数据库中导出。它包括了类似于 OQL 的说明性表达式,以利于超媒体模型从中抽取数据。

·超媒体模型:描述了要在 Web 上发布的信息内容、结构和相应的布局、显示。从三个方面支持超媒体建模:超媒体的组装、超媒体的导航、信息显示和内容的分离。它以对象作为基本的建模单位,Web 服务可以层次地分割成对象,一个复杂的对象可以描述一个页面、生成 Web 文档的 script 语言甚至一个网站。这样,一个网页可以由不同的对象组装而成,其中每一个对象都可以作为可重用的单元。每一个对象都有一个逻辑标识,通过对象标识的引用可以建模超媒体的导航结构。模板对象可以用来定义一组对象的公共布局,通过模板对象可以将内容和显示分开。

## 三、面向对象的 Web 建模语言(OOWML)

OOWML 是基于 XML 的标记语言,帮助 Web 应用开发者建模 Web 应用,有助于管理员管理 Web 之上的信息和动态生成的对象。由于 XML 本身是基于标准的标记语言技术,

用于创建可互换的、结构化的文档,作为基于 XML 的描述语言,OOWML 支持跨操作系统平台的对象模型的交换。OOWML 分为两部分:数据模型和超媒体模型。

### 1. 数据模型

OOWML 数据模型的基本元素是实体,它包含了相应的数据元素及其关系。实体具有属性,每一个属性都具有相应的名称和类型,和可选的 Value 来定义它的缺省的值。数据模型采用类似于统一建模语言(UML)中类图的概念<sup>[4]</sup>,实体可以以继承关系组织成层次结构。OOWML 数据模型支持实体间的“整体-部分”关系,集的约束(cardinality constraints)以及角色名(role name)的定义。

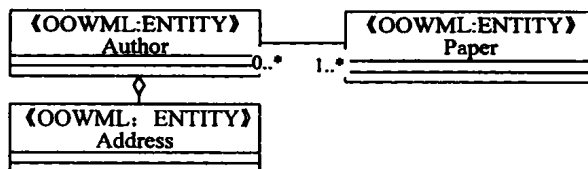


图1 数据模型的建模例子

例如,图1描述了一个数据模型的建模例子。其中的图形表示借用了统一建模语言(UML)中的符号。下面的 XML 代码是该例子的 OOWML 说明,包括三个实体: Author、Address 和 Paper, Author 和 Address 之间描述的是聚合关系, Author 和 Paper 之间是多对多的关联。其中,对于每一个 Paper,可以具有1到4个作者,每一个作者,可以有多个 Paper。

```

<OOWML:ENTITY ID="Author">
  <ATTRIBUTE NAME="Name" TYPE="String"/>
  <ATTRIBUTE NAME="birthDate" TYPE="Date"/>
  <ATTRIBUTE NAME="photo" TYPE="Image"/>
  <ATTRIBUTE NAME="biographicInfo" TYPE="Text"/>
  <ATTRIBUTE NAME="thePaper" TYPE="Paper" MIN="1" MAX="N"/>
  <OOWML:ENTITY NAME="address" MIN="1" MAX="1"/>
  <ATTRIBUTE NAME="street" TYPE="String"/>
  <ATTRIBUTE NAME="city" TYPE="String"/>
  <ATTRIBUTE NAME="country" TYPE="String"/>
  <ATTRIBUTE NAME="postcode" TYPE="String"/>
</OOWML:ENTITY>
</OOWML:ENTITY>
<OOWML:ENTITY ID="Paper">
  <ATTRIBUTE NAME="title" TYPE="String"/>
  <ATTRIBUTE NAME="abstract" TYPE="String"/>
  <ATTRIBUTE NAME="keywords" TYPE="Enumeration"/>
  <ATTRIBUTE NAME="theDate" TYPE="Date"/>
  <ATTRIBUTE NAME="publisher" TYPE="String"/>
  <ATTRIBUTE NAME="theAuthor" TYPE="Author" MIN="1" MAX="4"/>
</OOWML:ENTITY>
  
```

### 2. 超媒体模型

OOWML 超媒体模型使用对象作为统一的建模概念,这样,Web 应用可以层次地分割成对象。在较高的层次上,对象可以描述一个页面、生成 Web 文档的 script 语言甚至一个网站,在较低的层次上,对象可以描述粒度较小的资源,如表、链接等。对象描述了超媒体模型的最基本的建模实体,它由一组特定的特性构成,如对象标识、目的文件、MIMI 特性、对象引用等。对象也可以具有 HTML 属性,例如,一个表对象具有边界、单元格等属性。Web 应用的实施需要每一个对象实施自己的从状态信息到标准 Web 表示的映射服务,这种映射并不限于 HTML,也可以映射到 CSS 和 XSL。面向对象的 Web 模型支持对象之间的引用、组装和继承。对象引用建模了 Web 服务实体之间的关联关系,在实施模型中通常被映射为超媒体之间的链接;对象组装可以将 Web 服务描述为层次的组装关系,父对象由其下一层的子对象构成,处于最底层的通

常是原子对象,如文本、简单的图像等;对象之间的继承关系支持 Web 服务中的重用和共享机制,新的对象可以自然地加入已有的模型中,并继承已有对象的特性。

(1)OOWML 超媒体模型的基本元素 有对象、对象模板、引用等。对象描述了 Web 建模的最基本的建模实体。它由一组特定的属性构成,它都由名-值对表示。每一个对象都有下列中的一些属性:

- 对象标识:对象标识可以直接赋值或由编译器自动生成,逻辑上可以用对象标识来引用一个对象。

- 目标文件:目的文件特性控制了对象要创建的目标及其位置,存在于表示网页的对象,如:某个对象要生成 index.html,可以描述为 DST="/file:index.html"。

- MIMI 特性: OOWML 可管理任何 MIMI 类型的对象,如多媒体数据、图像、交互的 script 等。它们都有一个 MIMI 特性,描述了它的信息类型,缺省的类型是"/text/html"。

对象模板包含了没有绑定值的可变部分,它强调了对象的重用。要重用对象,必须先对对象模板进行初始化。

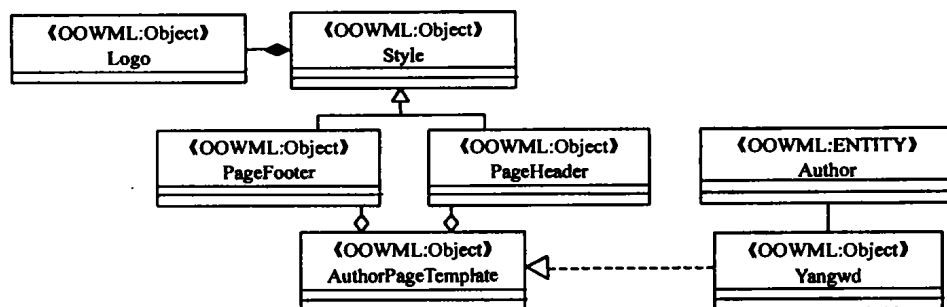


图2 超媒体模型的例子

上面的例子描述了我们为介绍“数字图书馆关键技术研究”项目建立的网站模型的一个片断,它包括7个对象:Style、Logo、PageFooter、PageHeader、AuthorPageTemplate、Yangwd、Author。Style 对象的描述如下:

```
<OOWML:OBJECT ID="Style">
...
<ATTRIBUTE NAME="font-style" VALUE=' face="Verdana,
Helvetica, Arial" size="2"/>
<ATTRIBUTE NAME="background" VALUE="#F0F0E1"/>
<OOWML:OBJECT ID="Logo"/>
<ATTRIBUTE NAME="dir" SRC=" theDirectory/loGoPic-
ture.gif"/>
<ATTRIBUTE NAME="type" MIMI="image/gif"/>
</OOWML:OBJECT>
</OOWML:OBJECT>
```

Style 对象描述了网站风格的相关信息,如字体、背景颜色等,将网站风格的一些相关信息封装在对象中有助于保持网站风格的一致和系统的维护。Style 对象与 Logo 对象之间的关系是强聚合关系,强聚合关系是一种更紧密的聚合关系,要求整体与部分具有相同的生命周期<sup>[4]</sup>,在 OOWML 中使用嵌入的描述方式。弱聚合关系是较为松散的整体-部分关系<sup>[5]</sup>,不要求整体与部分具有相同的生命周期,表示部分的对象可以离开整体独立存在,在 OOWML 中通过属性中值的引用描述。下面的例子描述了 AuthorPageTemplate 对象和 PageFooter 对象、PageHeader 对象之间的弱聚合关系,对象的表示信息在“content”属性中描述。

```
<OOWML:OBJECT id="AuthorPageTemplate">
<ATTRIBUTE NAME="myPageName" DST="index.html"/>
<ATTRIBUTE NAME="theFileType" MIMI="text/html"/>
<ATTRIBUTE NAME=" theFooter " VALUE="
PageFooter"/>
<ATTRIBUTE NAME=" theHeader " VALUE="
```

对象可以引用其它对象建模 Web 应用资源之间的关联和导航结构。有时,引用被解析为对象之间的链接。

(2)OOWML 超媒体模型的建立 Web 超媒体的应用通常是以基于文件的资源进行开发和交付,缺乏抽象的概念和模型,难以支持系统的分析和设计,系统的维护也非常困难。OOWML 超媒体模型以对象作为建模的基本概念,抽象地描述了 Web 服务对象的结构和表示信息,能够较好地支持 Web 服务的分析和设计,并自始至终提供一致的模型。软件重用是支持有效的软件开发和系统维护的重要手段,它是 OOWML 超媒体模型的一个重要目标。由于对象可以以任意的粒度描述 Web 服务,包括网页中的一个构件、网页,甚至一个网站,通过对象的组装和继承,OOWML 可以较好地支持超媒体模型的重用,也可以通过设计模式<sup>[6]</sup>支持 Web 服务的设计应用。例如,通过重用网站公共的布局信息和显示信息,可以保证网站具有一致的风格,在需要对这些信息修改时,也利于网页之间的一致性维护。

```
PageHeader"/>
<ATTRIBUTE NAME="content">
...
About the detailed information for the
< OOWML:VAR NAME="author"/>
is listed below
< OOWML:VAR NAME="descriptions"/>
</ATTRIBUTE>
</OOWML:OBJECT>
```

“AuthorPageTemplate”描述了关于所有作者的公共的布局和显示信息。对象模板包括了可变的,它是指没有被赋值的变量。对象模板通常用来建立公共的布局信息以利于系统的维护和风格的一致性。对于对象模板的重用通过继承关系实现,对象对模板对象的继承实际上是对模板的具体实现,模板的继承首先要对模板进行初始化,在初始化过程中要进行变量到值的绑定。下面的 OOWML 代码描述了“Yangwd”对象对模板“AuthorPageTemplate”的继承。

```
<OOWML:OBJECT ID=" Yangwd" SUPERTEMPLATE=" AuthorPageTemplate"ENTITY=" Author">
<INCLUDEALL KEY="Name" VALUE="yangwd"/>
<ATTRIBUTE NAME="myintrest" LINKTO="myintrestObject"/>
<ATTRIBUTE NAME="content">
<OOWML:OBJECT DST=" Yangwd. Name">
</OOWML:OBJECT>
<OOWML:OBJECT DST=" Yangwd. biographicInfo">
He is a postdoctor </OOWML:OBJECT>
</ATTRIBUTE>
</OOWML:OBJECT>
```

上述例子还表明了“Yangwd”对象如何定义了数据模型中实体“Author”的一个视图,通过 ENTITY="Author"表示。对实体属性的选择使用 INCLUDE 元素说明。这种说明必须包括两部分:1)说明引用了数据模型中的哪个实体。2)选择了实体中的哪些属性,INCLUDEALL 缺省的选择实体中的

所有属性。一个对象可以从多个实体中导出信息,同样,多个对象可以从一个实体中导出信息。可以看出,通过将超媒体模型和数据模型分开,对同一个数据模型可以定义多个超媒体视图;在超媒体模型中,对于同样的内容,又可以定义多种表示的视图;这种层次模型有利于 Web 应用的重用,同样对于 Web 应用的维护也非常重要。

#### 四、开放的 OOWML 实施

可扩展标记语言(XML)<sup>[3]</sup>支持用户自己定义特定的语义,XML 解析器广泛适用于大部分平台(如 Unix 和 WindowsNTXML)。OOWML 编译器用 Java 语言实施,并基于已有的 XML 解析器,支持平台的可移植性,同时支持开发者描述和互操作 Web 应用开发模型。

OOWML 编译器输入以 oowml 为扩展名的文本文件,根据 OOWML 模型自动生成相应的基于文件的 Web 表示(如:HTML)。它从根对象开始,从复杂对象到原子对象自顶向下扫描文档,生成相应的结果树以分析对象之间的关系。解析之后,“content”属性生成相应的表示输出。最后,编译器根据对象的标识和它的 DST 属性创建相应的目标资源。

**结论** 由于特定的基于文件的 Web 服务结构以及缺乏系统化的方法、模型和工具的支持,使得复杂 Web 应用的开发、维护和管理十分困难。较早的网站开发工具(如微软的 FrontPage 等)仅支持网页的编辑;近来研究领域出现的开发工具和方法,例如 OOHDM、RMM 等<sup>[2,4-6]</sup>,结合了传统的软件开发方法(面向对象或结构化方法),较好地支持了复杂 Web 应用的开发,但模型非常复杂,又较少考虑 Web 应用的管理和维护。

与 OOHDM 相比,OOWML 更为简单,同时采用了基于 XML 的描述和实施,具有 XML 所拥有的优点,利于 OOWML 开发模型的互操作。与 WebComposition 相比,OOWML 采用两层体系结构模型,支持同一数据模型的多超媒体视图以及同一内容的多视图显示,较好地改善了 Web 应用的可管理性、可维护性和可扩充性。

我们开发 OOWML 及其编译器,正应用于复旦大学计算机系数据库组承担的国家重点项目“数字图书馆关键技术研究”中的相关网站建模、数据发布和站点维护。为了更全面地支持 Web 应用开发,进一步的工作有:基于 XML 技术,集成

特定于 Web 应用领域的设计模式<sup>[9,10]</sup>,以支持多层次的软件重用;开发基于 Web 的可重用构件库;利用 Rational Rose 的可扩充接口 REI,构造与 Rational Rose 集成的 Web 应用开发环境。

#### 参考文献

- 1 Gellersen H W, Wicke R, Gaedke M. WebComposition: An Object-Oriented Support System for the Web Engineering Lifecycle. Computer Networks and ISDN System, 1997, 29: 429~437
- 2 Barta R, Schranz M W. JESSICA - An Object-Oriented Hypermedia Publishing Processor. Computer Networks and ISDN Systems, 1998, (April): 239~249
- 3 World Wide Web Consortium. Extensible Markup Language (XML 1.0) Specification, tech. Report. <http://www.w3.org/TR/REC-xml>.
- 4 Isakowitz T, Stohr E A, Balasubramanian P. RMM: A methodology for Structured Hypermedia Design. Communications of the ACM, 1995, 38(8)
- 5 Schwabe D, Rossi G. An Object-oriented Approach to Web-Based Applications Design. Theory and Practice of Object Systems, 1998, 4(4): 207~225
- 6 Gaedke M, Graef G. Development and Evolution of Web-Applications using the WebComposition Process Model. In: Intl. Workshop on Web Engineering at the 9th Intl. World-Wide Web Conf. (www), May 2000
- 7 Johnson R E, Foote B. Designing Reusable Class, Journal of Object-Oriented Programming (JOOP), 1998, 1(2)
- 8 Rational Corporation, UML 1.1 Documents. <http://www.rational.com>.
- 9 German D, Cowan D. Towards a Unified Catalog of Hypermedia Design Patterns. In: 33rd Hawaii Intl. Conf. on System Sciences (HICSS 2000), Maui, Hawaii, Jan. 2000
- 10 Gamma E, Helm R, Johnson R, Vlissides J. Design Patterns: Elements of Reusable Object-Oriented Software, Addison-Wesley, 1994
- 11 Atzeni P, Mecca G, Merialdo P. To Weave the Web. In: Proc. 23rd Conf. on Very Large Databases, Aug. 25~29, 1997, Athens, Greece, pages 206~215

(上接第114页)

#### 4.3 创建其它服务单元组建分布式服务群

类似 Chart-Server 的创建,我们可以生成很多可以被不同的客户调用的服务单元,再将这些服务单元分布到拥有类似 ORB 这样的网络环境中去,对每一种服务生成完备的接口和功能文档。在该系统中完全可以由其他软件开发人员开发服务程序,然后组装到分布式的网络环境中,再由另外一方来开发客户端的应用。

**结束语** 在分布式服务群基础上的信息化,比传统的信息管理将具有更方便的维护和更好的扩展性,同时也减轻了上层应用开发的复杂性和层次混乱的情况。但该模型还需在

网络的安全性及负载的均衡性方面进一步完善。

#### 参考文献

- 1 Otte R, Patrick P, Roy M. CORBA 教程:公共对象请求代理体系结构. 清华大学出版社, 1999
- 2 Wu Jie. Distributed System Design. 机械工业出版社, 2001
- 3 Mandell S L, Sakthivel S. Computers and Information Processing. 机械工业出版社, 1999
- 4 Abernethy R. COM/DCOM 技术内幕. 电子工业出版社, 2000
- 5 Pressman R S. Software Engineering - a Practitioner's Approach. 机械工业出版社, 1999
- 6 李维. Delphi 5. x 分布式多层应用系统篇. 机械工业出版社, 2000