

# 第 1 章 Web 程序设计基础



随着互联网的普及和广泛应用，在软件开发中，基于 B/S 三层结构的 Web 应用框架已经成为市场的主流，传统 C/S 结构的程序风光不再。上述情形只是出现在软件公司中，在高等学校，专门讲授 Web 开发的课程还很少，Web 程序设计还处于起步阶段。

本章首先介绍计算模式与程序设计模式的演变，对不同模式下的计算机程序体系结构进行对比，然后给出 Web 应用（程序）的概念、体系结构，讲解 Web 的工作原理。最后给出 Web 中的主要概念，为后续章节的学习给出一个框架和知识铺垫。



- 计算模式、软件设计与开发模式的演化
- Web 中的 B/S 三层体系结构及其工作原理
- Web 中的相关概念，包括 Java 技术、XML 技术、Web 服务等
- Web 新进展、Web2.0、语义 Web 等

## 1.1 计算模式与程序设计模式的演变

1946 年，第一台电子计算机“爱尼亚克”（ENIAC）在美国宾夕法尼亚大学莫尔电子工程学院诞生。这种计算技术的革命，透出了数字信息时代的第一缕曙光，开启了计算机程序设计的历史。进入 20 世纪 90 年代以后，随着 Internet 技术的不断发展，特别是 Web 的出现，对计算机的计算模式、软件开发模式、应用模式都产生了重要的影响。

在 Web 中，B/S 三层结构的提出，突破了传统的程序单机运行模式和基于局域网的 C/S 运行模式，突破了局域网的约束，将计算机应用分布到了整个互联网中，使计算机程序的设计和应用模式都发生了革命性的变化，进而推动了 Web 应用的快速发展。

### 1.1.1 单机运行与集中式计算模式

在计算机诞生和应用的初期，计算所需要的数据和程序都是集中在一台计算机上进行的，称为集中式计算。随着网络的发展，这种集中式计算往往形成一种由大型机和多个与之相连的终端组成的网络结构。当支持大量用户时，大型机自顶向下的维护和管理方式显示出集中式处理的优越性。它具有安全性好、可靠性高、计算能力和数据存储能力强以及系统维护和管理费用较低等优点。但是它也存在着一些明显的缺点，如：大型机的初始投资较大、可移植性差、资源利用率低以及网络负载大等。

此时的计算机程序采用了源程序、编译、连接到运行的这样一个简单的开发过程，最终

形成一个可在计算机操作系统上运行的可执行文件。这个可执行文件被安装在计算机上，它是早期程序主要的应用模式，同时也是 20 世纪 80 年代 C/S 计算模式下的主要应用形式。单机环境下的程序开发过程如图 1-1 所示。

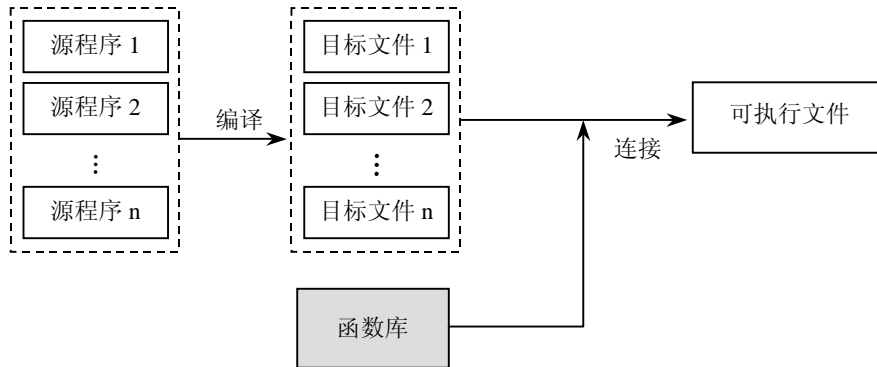


图 1-1 早期的编译型计算机程序开发过程

编程人员利用开发工具（如 C、Fortran）来编写程序，即源程序。然后对源程序进行编译、连接操作，最终形成一个可执行程序，即一个 exe 文件。这种开发模式一直影响到 C/S 计算模式，直到 Web 的出现，一种基于 Web 服务的程序开发和运行模式开始出现和被广泛地应用，基于 Web 的软件开发开始成为程序设计和应用的主流。

### 1.1.2 客户/服务器（C/S）计算模式

20 世纪 80 年代，随着微型计算机和网络的发展，数据和应用逐渐转向了分布式，即数据和应用程序跨越多个节点机，形成了新的计算模式，这就是客户/服务器（Client/Server, C/S）计算模式。C/S 模式是一种典型的两层计算模式，它将应用一分为二：前端是客户机，一般使用微型计算机，几乎所有的应用逻辑都在客户端进行和表达，客户机完成与用户的交互任务，具有强壮的数据操纵和事务处理能力。后端是服务器，可以使用各种类型的主机，服务器负责数据管理，提供数据库的查询和管理、大规模的计算等服务。

C/S 计算模式具有以下几个方面的优点：通过异种平台集成，能够协调现有的各种 IT 基础结构；分布式管理；能充分发挥客户端 PC 的处理能力，安全、稳定、速度快，且可脱机操作。但随着应用规模的日益扩大，应用程序的复杂程度不断提高，C/S 结构逐渐暴露出许多的缺点和不足，主要包括：它必须在客户端安装大量的应用程序（客户端软件），开发成本较高，移植困难，用户界面风格不统一，使用繁杂，不利于推广使用，维护复杂，升级麻烦，信息内容和形式单一，新技术不能轻易应用等。

### 1.1.3 浏览器/服务器（B/S）计算模式

在 Web 出现以前，计算机的应用模式经历了单机应用到网络应用两个阶段，这些不同的计算模式有各自的优点和不足。Web 的出现使得一种围绕 Web 服务的计算模式成为当前计算机应用的主流模式，并推动了软件开发、软件应用、应用集成方式上的重大改变。

#### 1. 互联网及 Web 的发展

在互联网的发展历史上，有几个历史性的事件和时刻。首先是 1969 年美国国防部高级研

究计划署 ARPA 资助的一个有关广域网络的项目，即阿帕网（ARPANet），它为后来 Internet 的出现奠定了基础。1969 年 11 月 21 日中午，6 名科学家聚会美国加利福尼亚大学洛杉矶分校的计算机实验室，观看这里的一台计算机与远在千里之外的斯坦福研究所的另一台计算机连通。这是一个历史性的时刻，正像 20 年后《时代》周刊的评论：这些研究者根本没有想到，他们不只是连接了两台计算机，而是宣告了网络世界的到来。

在接下来的短短 30 多年的时间里，从 1969 年到 1984 年的这个时期，是 Internet 的提出、研究和试验阶段。1984 年到 1992 年期间，是 Internet 的实用发展阶段。特别是 1989 年，这一年出现的万维网技术给 Internet 赋予了强大的生命力，它是互联网历史上划时代的水分岭，万维网将 Internet 带入了一个崭新的时代。1992 年以后，Internet 开始进入它的商业化发展阶段，万维网的出现，使 Internet 的使用更简单、更方便，Internet 用户开始向全世界扩展，开创了 Internet 发展的新时期。

## 2. B/S 计算模式

客户/服务器（C/S）模式表现出了许多不足，特别是它的胖客户机和对局域网的依赖，已经不能适应 Web 的发展。人们需要利用互联网，将应用分布到整个 Web 中，而不是局限于企业局域网内部，这就导致了一种更加灵活的多级分布式计算模式，即浏览器/服务器模式（Browser/Server, B/S）的产生和发展。

浏览器/服务器（B/S）计算模式是一种基于 Web 的协同计算，是一种三层架构瘦客户机/服务器计算模式。第一层为客户端表示层，与 C/S 结构中的“胖”客户端不同，三层架构中的客户层只保留一个 Web 浏览器，不存放任何应用程序，其运行代码可以从位于第二层的 Web 服务器下载到本地的浏览器中执行，几乎不需要任何管理工作，是一种“瘦”客户机。第二层是应用服务器层，由一台或多台 Web 服务器组成，处理应用中的所有业务逻辑、对数据库的访问等工作。该层具有良好的可扩充性，程序的部署和管理主要在 Web 服务器上进行，相对于 C/S 而言无论是工作的复杂性还是工作量都大大减少。第三层是数据中心层，安装数据库服务器，负责整个应用中的数据管理。

B/S 计算模式与传统的 C/S 结构相比体现了集中式计算的优越性：具有良好的开放性，利用单一的访问点，用户可以在任何地点使用系统；用户可以跨平台以相同的浏览器界面访问系统；因为在客户端只需要安装浏览器，基本上取消了客户端的维护工作，有效地减少了整个系统的运行和维护成本。

## 3. Web 应用系统

和传统的单机运行模式不同，Web 下的程序和传统的可执行程序有着太多的区别。一般意义下，根据操作系统的不同，程序有 Windows 程序、UNIX 程序等，一个在 Windows 下运行的应用程序，在 UNIX 或 Linux 下是不能直接运行的，这就是程序的可移植性问题。

随着 Windows 操作系统的出现，应用程序在外观上和操作上有了一个基本的模式，那就是基于窗口的程序。每个程序都有一个主窗口，窗口上有标题栏、菜单栏、工具栏等，用户通过执行菜单命令或工具按钮来执行相应的操作。这就是传统的应用程序给我们的感受。

但是，在 Web 中，Web 程序或者说 Web 应用，和上述我们对程序的印象完全不同。一个 Web 应用是由组织在一个文件夹（Web 应用或站点主目录）下的大量的文件和子文件夹组成的，这些文件就是我们看到的一个个网页，它们之间通过超链接来表达业务逻辑，实现传统程序中的菜单功能。这些页面不再是传统意义上的一个个源程序经过编译、链接而成的，它们不再需要安装在用户的计算机上，而是被部署在一台称为 Web 服务器的计算机上，用户通过

Internet 运行 Web 服务器上的程序，为互联网中的所有用户提供 Web 服务。

#### 1.1.4 软件设计与开发模式的演化

从计算机诞生到现在，计算机硬件技术在发展的同时，计算机软件也在悄悄地发生变化。这种变化不仅表现在计算机软件的内涵、计算机软件的应用方式上，同时也表现在软件的设计模式和开发方法上。

站在软件开发人员的角度，往往希望软件开发能够满足对于开发效率、可靠性、易维护性、易管理等多方面的更高要求。无论在计算机发展的哪一个时期，这都是软件开发人员永恒的目标。可以把这种目标方法的实现分成以下几个阶段：

(1) 面向机器语言的开发模式。在计算机诞生的初期，需要根据不同平台的机器语言来开发代码。这一阶段，程序规模很小，谈不上系统的方法。编程人员使用机器语言和汇编语言编写代码，效率很低，代码几乎没有可移植性。

(2) 软件的生命周期开发模式。又称软件开发瀑布模型，是面向功能或过程的软件开发方法。它将软件开发分成几个阶段：①用户需求分析，开发人员和业务人员交流，制定用户需求说明书；②系统设计，开发人员根据需求说明书进行系统设计，制定系统设计报告；③系统开发，根据系统设计报告，进行系统编码；④系统测试，系统实现后双方组织人员进行测试；⑤运行与维护，系统测试结束后，便进入系统的运行、维护期。

从理论上讲，软件开发生命周期模式是非常科学的，但是利用生命周期模式开发系统基于两个假设：①用户能够清楚地、完整地提供系统要求；②开发者能完整地、严格地理解和定义要求。但在实际开发中，这两个假设是很难满足的。因为，在开发初期，用户很难清楚地描述系统需求，或者系统需求将来可能发生较大变化。其次，开发人员和业务人员在交流时可能存在理解上的不一致，其结果是系统开发完毕后，不能很好地满足用户需求。生命周期模式是封闭式的，缺少灵活性，特别是在用户需求定义方面。

面向过程的生命周期法主要流行于 20 世纪 80 年代，当时的编程语言主要是 C、Fortran 等过程式程序设计语言。相对于早期的汇编程序，高级语言的出现已经是一种很大的进步。它极大地提高了编程效率，提高了程序的可移植性。直到现在，这种思想一直还对软件设计和开发人员有着很深的影响。

(3) 原型法开发模式。和生命周期模式不同，原型法要求经过对用户需求的简单快速分析，利用高级开发工具及环境，快速完成原型系统的设计和实现，提供给用户评价。一个原型系统就是系统的一个可运行的早期版本，它反映了最终系统的部分重要特征，在评价过程中，开发人员不断从用户那里得到反馈信息，修正原型的用户需求定义，进而对原型系统作相应改进，逐步减少分析与交互过程中的误解，弥补遗漏，从而提高最终系统的质量。

原型法的核心是用交互的、快速建立起来的原型取代形式的、不易修改的大块的规格说明，用户通过在计算机上实际运行和试用原型而向开发者提供真实的反馈意见。原型法开发模式的实现得益于面向对象的语言（Smalltalk、C++、Java 等）和可视化的第四代开发工具的出现。从宏观上讲，原型法比生命周期更实用，但是在每一个原型的设计和开发过程中，都离不开生命周期的科学思想。

在软件工程的实践中，生命周期法和原型法的有效结合表现出了强大的生命力和可操作性。这种结合就是，整个软件的开发表现为一个个原型的向前推进，在每一个原型的内部，又

是按照生命周期的思想来设计。

(4) 面向组件 (Component) 的模式。随着软件开发规模的扩大, 在涉及分布式、异构等复杂特征的环境中, 代码级别的重用性差、可维护性差、效率低的弱点是不可逾越的, 因此人们以架构运行环境 (如 .Net、J2EE 等) 来提供完善的支撑平台, 从而把开发者解放出来, 更专注于业务核心的开发。而这些业务功能 (Business Function) 以组件的形式 (DCOM、EJB 等) 发布运行在架构运行环境中, 软件开发的复用模式也上升到业务组件的级别。

(5) 面向方面编程 (Aspect Oriented Programming, AOP) 模式。AOP 是 OOP 的延续, 设计模式孜孜不倦追求的是调用者和被调用者之间的解耦, AOP 可以说也是这种目标的一种实现。AOP 使原本复杂的调用与被调用和错综复杂的耦合关系变得清晰, 使程序的整体架构保持高内聚, 低耦合, 这对于一个大型复杂系统来说是非常重要的。

(6) 面向服务 (SOA) 的模式。当软件的使用范围扩展到更广阔的范围, 往往会面对更加复杂的 IT 环境和更加灵活多变的需求。服务 (Service) 的概念出现了, 人们将应用 (Application) 以业务服务 (Business Service) 的形式公布出来供别人使用, 而完全不需要考虑这些业务服务运行在哪一个架构体系上, 因为所有的服务都讲着同样的语言。

SOA 同样也强调重用 (Reuse), 但是相对于传统的代码重用、对象重用和部件重用, SOA 的重用粒度更粗。SOA 的核心体现在企业应用或者业务功能上的“重用”和“互操作”, SOA 的重用在于业务级的应用, 即服务的重用, 而不再把 IT 与业务对立起来, 这可以被视为在 IT 驱动业务的方向上迈出的重要一步。

## 1.2 Web 及其工作原理

1990 年, 瑞士日内瓦世界上最大的粒子物理研究实验室欧洲核子研究中心 CERN (the European Organization for Nuclear Research) 提出了 World Wide Web (WWW) 的概念, 它是 Internet 技术、超文本技术和多媒体技术相结合的产物。当时, 核物理的研究是分散在不同国家进行的, 各地的研究人员通过计算机网络和 Internet 进行学术交流。在 Internet 中信息交流还没有一种统一的手段, 因此根据交流的信息不同 (如图片、文字等) 需要调用不同的 Internet 服务, 很不方便。1989 年 3 月, CERN 的 Tim Berners-Lee 开发了一个超级文本系统, 1990 年底, 第一个基于字符界面的 Web 客户浏览程序开发成功, 1991 年 3 月, 客户浏览程序开始在 Internet 上运行, 1991 年底 CERN 向高能物理学界宣布了 Web 服务。

### 1.2.1 什么是 Web

什么是 Web 呢? 从万维网诞生起, 人们并没有给它一个确切的定义。我们可以从 Internet 的构成和服务来理解 Web。

Internet 是一个网络上的网络, 或者说是一个全球范围的网间网。在 Internet 中, 分布了成千上万的无以计数的计算机, 这些计算机扮演的角色和所起的作用不同。有的计算机可以收发用户的电子邮件, 有的可以为用户传输文件, 有的负责对域名进行解析, 更多的机器则用于组织并展示本网络的信息资源, 方便用户的获取。所有这些承担服务任务的计算机统称为服务器。根据服务的特点来区分, 又分成邮件服务器、文件传输服务器、DNS 服务器、Web 服务器等。

所谓 Web 服务器, 就是将本地的信息用超级文本组织, 向用户提供在 Internet 上进行信息

浏览服务的计算机。因此，Web 或者说 World Wide Web，是由 Internet 中称为 Web 信息服务器的计算机组成的，它们由那些希望通过 Internet 发布信息的机构提供并管理。在 Web 世界里，每一个 Web 服务器除了提供自己独特的信息服务外，还可以用超链接指向其他的 Web 服务器。那些 Web 服务器又可以指向更多的 Web 服务器，这样一个全球范围的由 Web 服务器组成的 World Wide Web（万维网）就形成了。

### 1.2.2 Web 的工作原理

万维网的运行是一种典型的浏览器/服务器（Browser/Server, B/S）模式。典型的 B/S 体系结构将计算机应用分成三个层次，即：客户端浏览器层、Web 服务器层和数据库服务器层。B/S 体系结构有许多优点，它简化了客户端的维护，所有的应用逻辑都是在 Web 服务器上配置的。更主要的是，B/S 应用模式突破了传统的 C/S 模式中计算机应用对局域网的局限，从而将计算机应用分布到整个互联网中，用户可以在任何地方登录 Web 服务器，按照用户角色，执行自己的业务程序。

在万维网中，通过 HTTP 协议实现客户端（浏览器）和 Web 服务器的信息交换，Web 的基本工作原理如图 1-2 所示。

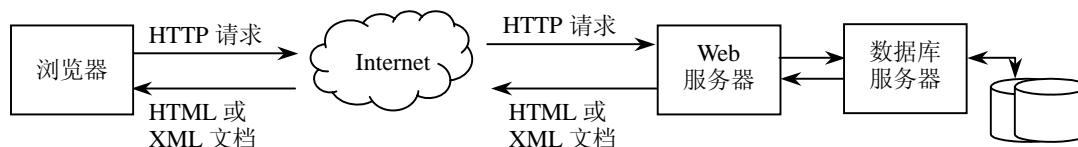


图 1-2 Web 的工作原理

在浏览器地址栏中，用户输入要访问的网页网址 URL（http://网址/路径/文件名.扩展名），向 Web 服务器提出 HTTP 请求。Web 服务器根据 URL 中指定的网址、路径和网页文件，调出相应的 HTML、XML 文档或 JSP、ASP 文件，根据文档的类型，Web 服务器决定是否执行文档中的脚本程序，还是直接将网页文件传送到客户端。

一般情况下，所有的 Web 应用几乎都要用到数据库管理数据。对数据库的管理和操作都是通过数据库服务器完成的，这些程序代码被以服务器端脚本的方式编写在 ASP、JSP 等页面中，负责和数据库服务器建立连接并完成必要的数据库查询、插入、删除、更新等数据库操作，然后利用获得的数据产生一个新的包含动态数据的 HTML 或 XML 文档，并将其发送给客户端 Web 浏览器。最后由 Web 浏览器解释该文档，在浏览器窗口中显示给用户。

### 1.2.3 Web 浏览器

Web 浏览器（browser）就是前面经常提到的 Web 客户端程序，用户要浏览 Web 页面必须在本地计算机上安装浏览器软件。通过在浏览器地址栏中输入 URL 资源地址，将 Web 服务器中特定的网页文件下载到客户端计算机中，并在浏览器打开。因此，从本质上讲，浏览器是一种特定格式的文档阅读器，它能够根据网页内容，对网页中的各种标记进行解释显示。此外，浏览器又是一种程序解释机，如果网页中包含客户端脚本程序，浏览器将执行这些客户端脚本代码，从而增强网页的交互性和动态效果。不同版本的浏览器都需要遵循 HTML 规范中定义

的标记集，同时为了便于脚本编程，每个浏览器程序本身也提供了相应的浏览器内置对象，类似于传统软件开发中的函数库及其标准库函数。

在 Web 发展初期，浏览器程序主要分成两类。一类为以 Lynx 为代表的基于字符的 Web 客户机程序，主要在不具备图形图像功能的计算机上使用。Lynx 是由美国堪萨斯大学的 Lou Montulli 研制的，同类的还包括 CERN 的 LineMode Browser。另一类是以 NCSA (National Center of Supercomputing Application) Mosaic 为代表的面向多媒体计算机的 Web 客户机程序，它可以在各种类型的小型机上运行，也可以在 IBM PC 机、Macintosh 机以及 UNIX 操作系统软件平台上运行。目前，使用较多的浏览器是微软的 IE (Internet Explorer) 浏览器。除此之外，一些新的浏览器产品也不断推向市场，如 Maxthon (傲游)、Firefox (火狐狸)、Opera 等。

#### 1.2.4 Web 服务器

上面已经多次提到 Web 服务器，那么什么是 Web 服务器呢？我们知道，万维网是由分布在 Internet 中的 Web 服务器组成的。在 Internet 中，Web 服务器有两个层面的含义，一是指安装了 Web 服务器的计算机，二是指 Web 服务器程序。所谓 Web 服务器程序，简单地讲就是一个服务程序，它仅仅需要监听合适的端口，建立连接，然后发送数据给客户端。服务器程序的开发总是和客户端软件的开发相辅相成的。

要使一台计算机成为一台 Web 服务器，一般需要服务器操作系统，如 UNIX、Windows Server 2003、Linux 等网络操作系统，并且还要安装专门的信息服务器程序，如 Windows 中的 Internet 信息服务器 IIS (Internet Information Server)、Apache/Tomcat 等。Web 服务器的种类在一定程度上决定了 Web 应用的开发环境。

## 1.3 Web 相关技术

在互联网中，Web 几乎成为 Internet 的代名词，它给我们的工作和生活带来了无限的便利。在 Web 领域，新的理念不断产生，相关技术层出不穷，这在推动 Web 技术发展的同时，也无形中提高了 Web 应用开发的复杂性和入门的难度。从宏观上，建立一个准确的 Web 相关技术整体框架对于 Web 技术的理解，以及未来的 Web 应用开发具有极其重要的意义。

#### 1.3.1 概念及术语

在 Web 中，新的概念、术语很多，随着 Web 应用的普及，这些本来是专业的概念和术语已经大众化了。下面从计算机专业的角度对 Web 中的一些常用概念进行简要介绍。

(1) 网站 (Web Site)。网站又称 Web 站点，是 Internet 中提供信息服务的机构，这些机构的计算机连接到 Internet 中，向用户提供 Web 服务。

从技术上讲，一个 Web 站点是由一个主目录和主目录下的文件夹和大量的网页文件构成的，这些网页文件通过超链接连接在一起，形成特定的应用逻辑，构成一个特定的 Web 应用。

(2) 超文本 (Hypertext)。超文本是一种文本显示与连接技术，可以对文本中的有关词汇或句子建立链接 (即超链接)，使其指向其他段落、文本或链接到其他文档。通过超链接，可以在文档之间、文档内部之间跳转，这种文本的组织方式与人们的思维方式和工作方式比较接近。

当超文本显示时，建立了链接的文本、图片通常以下划线、高亮等不同的方式显示，来

表明这些文本或图片对应一个超链接。当鼠标移过这些文字时，鼠标会变成手形，单击超链接文本或图片，可以转到相关的位置。

(3) 超级链接 (Hyperlink)。Web 页中当用户单击它时可以转到其他 Web 页或当前页面的其他地方的文字、图片等对象，即文本超链接和图片超链接。如果是文本超链接，超链接在 Web 页上往往带有下划线或增亮显示。当用户将鼠标指向一个超链接时，鼠标指针会改变为手的形状。

(4) Web 页 (Web Page)。Web 页是指 Web 服务器上的一个个超文本文件，或者是它们在浏览器上的显示屏幕。Web 页中往往包含指向其他 Web 页面的超级链接。

(5) 主页 (Home page)。用户在 Web 服务器上看到的第一个 Web 页，该 Web 页一般的名称为 default.htm 或 index.htm。首页中往往列出网站的信息目录，或指向其他站点的超链接，主页是一个网站的入口。当用户访问一个网站时，如果在 URL 中不指定特定的网页文件，则 Web 服务器将站点首页发送到客户端。

(6) 统一资源定位器 URL (Uniform Resource Locator)。统一资源定位器可以唯一标识一个 Web 页或 Internet 上其他资源的一个地址，它将 Internet 提供的各类服务统一编址，以便用户通过 Web 客户浏览程序进行信息查询。

URL 的一般形式为：

信息资源类型：//网址：端口号/文件路径/文件名？参数表

其中，信息服务类型主要包括 HTTP、FTP 等；网址即服务器的域名或 IP 地址，端口号对应一个特定的服务，默认端口号可以省略，例如 Web 服务的默认端口为 80，FTP 服务的默认端口为 21 等；文件路径为网页相对于主目录的相对路径；文件名是用户浏览器指定的要下载的网页文件；如果有参数，在文件名后面跟字符“？”列出参数名及实际参数值。

在浏览器地址栏 URL 中，默认端口号可以省略不写，如果不指定文件路径和文件名，则默认访问站点根目录下的首页文件，首页文件由 Web 服务器指定。例如，如果用户在浏览其中输入的 URL 为：<http://www.sdu.edu.cn>，则表明用户要下载域名为 [www.sdu.edu.cn](http://www.sdu.edu.cn) 的 Web 服务器中根目录下的首页文件。

(7) 端口 (Port)。端口是服务器使用的一个通道，在 OSI 参考模型的数据封装中使用，它可以使具有相同 IP 地址的服务器同时提供多种服务，运行在服务器上的各个服务程序将根据收到的数据段中的端口号来判断是否为自己的数据。例如，在一台计算机上同时提供 WWW 服务和 FTP 服务，WWW 服务使用端口 80，FTP 服务使用端口 21 等。在通用资源定位器 URL 中，默认端口号可以省略不写。

另外，在一台计算机上，在同一个 IP 地址下，可以建立多个 Web 站点，站点之间用端口号来区分，访问不同的站点必须指定特定的端口号。

(8) 下载 (Download)。下载是指通过 Internet 将文件从 Web 服务器或 FTP 服务器端传输到本地计算机的过程。

(9) 上传 (Upload)。上传是指通过 Internet 将文件从本地计算机传输到 Web 服务器或 FTP 服务器计算机的过程。

### 1.3.2 Java 技术

Java 技术是 Sun Microsystems 于 1995 年推出的一种极富创造力的计算平台。狭义上讲，



Java 技术可以理解为 Java 语言, 广义上讲, Java 技术包括 Java 语言、Java 虚拟机以及 Java API 等。Java 技术为用户带来了无数令人兴奋的可能性, 它几乎使所有应用程序(包括游戏、工具及信息程序和服务)都能在任何计算机或设备上运行。Java 技术的多功能性、有效性、平台的可移植性以及安全性已经使它成为网络计算领域最完美的技术。今天, Java 技术已经无处不在, 从桌面 PC 到科学超级计算机和互联网, 从移动电话到移动手持设备, 从家庭游戏机到信用卡, 几乎在所有的网络和设备上都会看到 Java 技术的身影。

### 1. Java 语言环境

1995 年, Sun 推出了 Java, 但这只是一种语言, 而要想开发复杂的应用程序, 必须要有一个强大的开发库支持。经过一年的试用和改进, Sun 在 1996 年 1 月 23 日发布了 JDK1.0。这个版本包括了两部分: 运行环境(即 JRE)和开发环境(即 JDK)。在运行环境中包括了核心 API、集成 API、用户界面 API、发布技术、Java 虚拟机(JVM)五个部分。开发环境包括编译 Java 程序的编译器(即 javac)。在 JDK1.0 时代, JDK 除了 AWT(Abstract Window Toolkit, 抽象窗口工具包, 一种用于开发图形用户界面的 API)外, 其他的库并不完整。

Sun 在推出 JDK1.0 后, 在 1997 年 2 月 18 日发布了 JDK1.1。JDK1.1 相对于 JDK1.0 最大的改进就是为 JVM 增加了 JIT(即时编译)编译器。JIT 和传统的编译器不同, 传统的编译器是编译一条, 运行完后再将其扔掉, 而 JIT 会将经常用到的指令保存在内存中, 在下次调用时就不需要再编译了, 这样 JDK 在效率上有了非常大的提升。

随后一些著名的计算机公司纷纷购买了 Java 的使用权, IBM、Apple、DEC、Adobe、Silicon Graphics、HP、Oracle、Toshiba、Netscape 和 Microsoft 等大公司相继购买了 Java 的许可证。另外, 众多的软件开发商也开发了许多支持 Java 的软件产品。在以网络为中心的计算时代, 不支持 HTML 和 Java, 就意味着应用程序的应用范围只能限于同质的环境。

Java 的平台无关性对未来的计算模式产生了革命性的影响, 它是继 HTML 后, Internet 发展的又一个里程碑。

### 2. Java 的技术特征

在 Sun 的 Java 语言白皮书中, 说明 Java 语言有如下特征: 简单、面向对象、分布式、解释执行、健壮、安全、体系结构中立、可移植、高性能、多线程、动态性等。

(1) 简单(Simple)。主要体现在三个方面: ①Java 语言风格来源于 C++, 因此 C++ 程序员可以很快的上手; ②Java 摒弃了 C++ 中容易引发错误的地方, 如指针, 增加了内存管理等一些新的特色; ③Java 提供了丰富的类库, 使用户编程更加简单。

(2) 面向对象(Object-Oriented)。Java 是面向对象的语言, 摒弃了 C++ 中全局变量等与面向对象思想冲突的内容。

(3) 体系结构中立(Architecture Neutral)。一般情况下, 网络环境都是异构的, 如何使一个应用程序能够在不同硬件、不同操作系统平台的计算机上运行, 始终是一个难题。Java 将它的程序编译成一种结构中立的中间文件格式, 由 Java 虚拟机来解释执行这种中间代码。这使得 Java 应用程序可以在不同的处理器中执行, 现在几乎所有的主流计算机系统都能运行 Java。

(4) 解释执行(Interpreted)。Java 解释器能直接在任何机器上执行 Java 字节码(Byte Codes)。

(5) 可移植(Portable)。同体系结构无关的特性使 Java 程序可以在配备了 Java 虚拟机的

任何计算机系统上运行。另外，通过定义独立于平台的基本类型及其运算，Java 数据得以在任何硬件平台上保持一致。

(6) 分布式 (Distributed)。Java 程序的程序库可以很容易地与 HTTP 和 FTP 等 TCP/IP 协议配合，从而使 Java 程序可以凭借 URL 打开并访问网络对象，对程序员来讲，访问方式和访问本地文件系统几乎一样，这就为 Internet 等分布环境提供内容带来了方便。

(7) 安全性 (Security)。Java 是被设计用于网络和分布式环境的，安全性自然是一个重要的考虑因素。Java 的安全性可以从两个方面考虑：①内存的安全性，如摒弃了 C++ 中的指针，从而避免了非法内存操作和内存泄漏；②当用 Java 来创建浏览器内容时，语言功能和浏览器本身的功能结合，使它更安全。

### 3. Java 的发展

十多年来，Java 技术的发展总是日新月异，从奠定 Java 根基的 Java 开发包 JDK1.0，到今天的 JDK6，Java 为开发人员提供的标准类库越来越丰富，Java 技术取得了长足的进步。

从 JDK1.0 到 JDK1.1.8，JDK1.x 经过了 9 个小版本的发展，已经初具规模。1998 年 12 月 4 日，Sun 发布了 Java 历史上最重要的一个 JDK 版本：JDK1.2，这个版本标志着 Java 进入了 Java2 时代，进入 Java 的飞速发展时期。

在 Java2 时代，Sun 对 Java 进行了很多革命性的变化，Sun 将 JDK1.2 一分为三，Java 被分成了 J2EE (Java2 Platform, Enterprise Edition)、J2SE (Java2 Platform, Standard Edition) 和 J2ME (Java 2 Platform, Micro Edition)，分别面向企业级、桌面、嵌入式和移动计算等领域。这些革命性的变化一直沿用到现在，对 Java 的发展形成了深远的影响。

从 JDK1.2 开始，Sun 以平均两年一个版本的速度推出新的 JDK。2000 年 5 月 8 日，Sun 对 JDK1.2 进行了重大升级，推出了 JDK1.3。Sun 在 JDK1.3 中同样进行了大量的改进，主要表现在一些类库上（如数学运算、新的 Timer API 等），在 JNDI 接口方面增加了一些 DNS 的支持，增加了 JNI 的支持等。2002 年 2 月 13 日，Sun 发布了 JDK 历史上最为成熟的版本 JDK1.4。这次 Sun 将主要精力放到 Java 的性能上，使 JDK1.4 的性能有了质的飞跃。到 JDK1.4 为止，已经可以使用 Java 实现大多数的应用了。

虽然从 JDK1.4 开始，Java 的性能有了显著的提高，但 Java 又面临着另一个问题，那就是复杂。2004 年 10 月，Sun 发布了 JDK1.5，同时，Sun 将 JDK1.5 改名为 J2SE5.0。和 JDK1.4 不同，JDK1.4 的主题是性能，而 J2SE5.0 的主题是易用。

2006 年 4 月，Sun 推出 J2SE6.0 测试版，2006 年 12 月，代号为 Mustang (野马) 的 J2SE6.0 正式版推向市场，在性能、易用性方面得到了前所未有的提高。

### 1.3.3 XML 技术

可扩展标记语言 XML (eXtensible Markup Language) 是 Internet 上最具权威的数据表示和数据交换标准，它是国际标准化组织 (International Organization for Standardization, ISO) 的通用标记语言标准 (Standard for General Markup Language, SGML) 的一个简化子集。

#### 1. XML 的技术特征

XML 关注信息本身，是 Web 上表示结构化信息的一种标准文本格式。与传统的注重页面信息显示的超文本标记语言 (Hypertext Markup Language, HTML) 相比，关注于内容的 XML 具有以下诸多优点：良好的可扩展性，语言简单有效，可自行定义标记；内容与形式的分离，

主要刻画数据内容，不考虑显示效果；有严格的语法要求，便于分析和与数据库信息转换；便于传输，为纯文本形式，可通过 HTTP 协议直接传输，可跨越防火墙等。

XML 的出现和发展对于 Internet 产生了巨大的影响，如果说 Java 实现了代码的平台无关性，那么 XML 则实现了数据的平台无关性。今天，XML 已经逐渐成为整个 Web 的基本结构和未来各种发展的基础，由于 XML 能针对特定的应用定义自己的标记语言，这一特征使得 XML 可以在电子商务、政府部门、各行业领域提供各具特色的独立解决方案。同时，XML 作为一种通用的数据交换语言，已经成为业界的一种具有垄断性的标准，在跨平台跨系统数据交换方面拥有无可比拟的优势，其在企业级开发中所扮演的角色越来越重要。但是，和关系数据库拥有强大的存储和分析引擎不同，XML 只专注于数据的表示，这也使得 XML 在数据量急速膨胀的时候，如何有效地管理和使用 XML 成为了一件令人头痛的事情。

## 2. XML 相关技术标准

虽然 XML 标准本身相对简单，但与 XML 相关的标准却种类繁多，W3C 制定的相关标准就有 20 多个，采用 XML 制定的各种应用标准也很多。除了标准种类繁多外，标准之间通常还互相引用，特别是应用标准，它们的制定不仅仅使用的是 XML 标准本身，还常常用到了其他很多标准。在 XML 标准体系中，XML 相关标准可分为元语言标准、基础标准、应用标准三个层次。

XML 相关技术标准体系如图 1-3 所示。

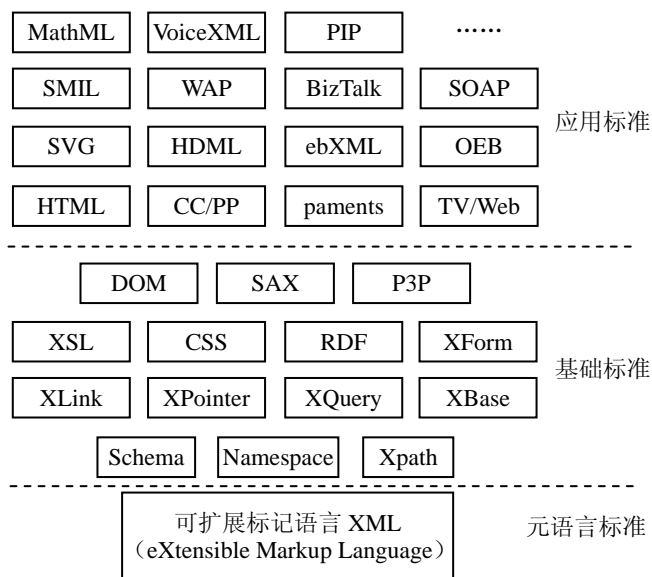


图 1-3 XML 技术标准体系

(1) 元语言标准 (Meta-Language)，描述的是用来描述标准的元语言，在 XML 标准体系中就是 XML 标准，是整个体系的核心，其他 XML 相关标准都是用它制定的或为其服务的。

(2) 基础标准 (Foundation Standards)，这一层次的标准是为 XML 的进一步实用化制定的标准，规定了采用 XML 制定标准时的一些公用特征、方法或规则。如：XML Schema 描述了更加严格的定义 XML 文档的方法，以便可以更自动化地处理 XML 文档；XML Namespace

用于保证 XML DTD 中名字的一致性，以便不同的 DTD 中的名字在需要时可以合并到一个文档中。

(3) 应用标准 (Application Standards)，以 XML 为基础制定的行业标准。比较常用的应用标准包括 SVG (有关矢量图形)、SMIL (有关多媒体同步显示)、MathML (有关数学公式符号) 等。在电子商务领域的常用应用标准有 Micropayments (W3C 制定)、BizTalk (Microsoft 发起的电子商务的 schema 库)、ebXML (联合国 UN/CEFACT 小组和 OASIS 共同发起)、PIP (由诸多 IT 业的巨子组成的一个标准化组织 RosettaNet 的应用网络标准) 等。

在 XML 技术标准体系中，主要的基础标准的功能划分如下：

XML Schema 描述了更加严格定义 XML 文档的方法，以便可以更自动地处理 XML 文档。XML Namespace 用于保证 XML DTD 中名字的一致性，以便不同的 DTD 中的名字在需要时可以合并到一个文档中。XPath 描述如何识别、选择、匹配 XML 文件中的各个构成元件，包括元素、属性、文字内容等。XPointer 和 XLink 标准，规定了有关定位、链接方面的内容。XQuery 的目的是为从 Web 文档中提取数据，提供一种灵活的查询机制。XSLT 则实现文档格式转换，主要是将 XML 转换为 HTML 格式进行显示。CSS 也是用来作为 XML 文档显示的样式标准。DOM 定义了一组与平台和语言无关的接口，以便程序和脚本能够动态访问和修改 XML 文档内容、结构及样式。

#### 1.3.4 Web 服务

Web 服务 (Web Service) 是在 XML 技术的基础上发展起来的，它是可以通过 Web 发布、查找和调用的自包含、自描述的模块化应用，其目标是将软件转化为一种可以通过 Web 订阅使用的服务。

Web 服务主要基于以下技术标准：

(1) 可扩展标记语言 XML。它是 Web 服务的驱动力。XML 语言不是一种编程语言或者 API，而是一种独立于平台的组织数据的方式。XML 的语法便于通过编程来处理文本数据，同时又便于为人们所理解。Web 服务使用 XML 作为标准，在网络设备之间进行通信。

(2) 简单对象访问协议 SOAP (Simple Object Access Protocol)。开发人员可以使用这种独立于平台的机制，远程调用分布式对象的方法。SOAP 消息的通信使用 XML 来描述对象、方法以及执行的参数。客户机和服务器都可以实现和使用 SOAP。

(3) Web 服务描述语言 WSDL (Web Service Description Language)。它从句法层面对 Web 服务的功能进行描述，包括四个不同的粒度：数据类型 (Data Type)、消息 (Message)、方法 (Operation) 和访问端口 (Port Type)。WSDL 只是提供了 Web 服务的接口描述，对服务的行为约束和属性描述缺乏进一步的支持。

(4) 语义 Web 服务标记语言 OWL-S。OWL-S 是语义 Web 服务标记语言的标准，它比 WSDL 更能向用户提供可理解的服务资源的描述形式，提高服务选取与推荐的准确性。语义 Web 服务的主要方法是利用本体 (Ontology) 来描述 Web 服务，然后通过这些带有语义信息的描述实现 Web 服务来实现服务的自动发现、调用和组合。语义 Web 和 Web 服务是语义 Web 服务的两大支撑技术。OWL-S 是连接两大技术的桥梁，目前对语义 Web 服务标记语言研究最重要的组织就是 DARPA 组织，其研究组 OWL Services Coalition 提出了语义 Web 服务标记语言 OWL-S。

(5) 通用描述发现和集成协议 UDDI (Universal Description, Discovery, and Integration)。该协议提供了一组公用的 SOAP API, 使得服务代理得以实现。UDDI 为发布服务的可用性和发现所需服务定义了一个标准接口(基于 SOAP 消息)。UDDI 实现将发布和发现服务的 SOAP 请求解释为用于基本数据存储的数据管理功能调用。

## 1.4 Web 发展趋势

Web, 这个由无以计数的超链接形成的网络世界, 在给我们每一个人的工作、学习、生活和娱乐带来了无穷的便利和全新的生命体验的同时, 新的技术、新的理念不断出现, 又让我们对 Web 的未来充满了无限的遐想。

### 1.4.1 Web 2.0

回想 Web 诞生之初, 我们面对一个个静态的网页, 但网页之间的超链接, 以及浏览器给我们的网络界面, 已经足够令我们兴奋不已。今天人们习惯把这个时期(2003 年以前的互联网模式)的互联网称为 Web1.0, 这是一个信息消费的时代, 人们通过浏览器获取信息。在 Web 1.0 时代, Netscape 脱颖而出, 成为互联网耀眼的新星, 它的浏览器, 把我们广大的普通用户带入了互联网。同时, Yahoo 提出了互联网黄页, Google 推出了深受欢迎的搜索服务, 它们为互联网的发展做出了巨大的贡献。

#### 1. Web 2.0 的概念

随着网络的发展, 网站的拥有者发现, 只有网民的参与, 才能持久地提高与保持网站的人气。从一开始出现的“论坛”到快速火热起来的“博客”, 互联网事实上已经逐渐开始了一种理念上的转变, 实践着从 Web 1.0 到 Web 2.0 的跨越。

2004 年 3 月, 在奥莱理(O'Reilly)媒体公司与 MediaLive 公司的一次头脑风暴会议上, O'Reilly 媒体公司负责在线出版及研究的副总裁戴尔·多尔蒂和 MediaLive 公司的克瑞格·克莱共同提出了 Web 2.0 的概念。2005 年 9 月 30 日, O'Reilly 媒体公司主席兼 CEO 提姆·奥莱理在其公司网站的个人栏目中发表文章“什么是 Web 2.0——下一代软件设计模式和商业模式”, 成为 Web 2.0 理念提出的一个重要里程碑。

关于 Web 2.0, 并没有一个统一的定义。互联网协会对 Web 2.0(互联网 2.0)的定义是: Web 2.0 是互联网的一次理念和思想体系的升级换代, 由原来的自上而下的由少数资源控制者集中控制主导的互联网体系转变为自下而上的由广大用户集体智慧和力量主导的互联网体系。Web 2.0 内在的动力来源是将互联网的主导权交还个人从而充分发掘了个人的积极性参与到体系中来, 广大个人所贡献的影响与智慧和个人联系形成的社群的影响就替代了原来少数人所控制和制造的影响, 从而极大地解放了个人的创作和贡献的潜能, 使得互联网的创造力上升到了新的量级。

如果说 Web 1.0 的主要特点在于用户通过浏览器获取信息, 那么 Web 2.0 则更注重用户的交互作用, 用户既是网站内容的消费者(浏览者), 也是网站内容的制造者。可见, 把 Web 2.0 理解成一个历史学的概念比技术性的概念更加准确, 对这个概念的梳理, 能帮助我们更好地把握互联网正在发生的技术与文化的变化。

## 2. Web 2.0 网站的特征

Web 2.0 是一种网站构建的理念，主要体现为以下特征：

(1) 网站平台应该有用户自己来定义的信息分类/分众标签功能和提供信息聚合 RSS (Rich Site Summary, 或 Really Simple Syndication) 功能, RSS 是一种描述和同步网站内容的格式, 是目前使用最广泛的 XML 应用。

(2) 网站内容应该主要来源于用户原创内容 (User Generated Content, UGC), 即此网站的内容和信息更多地在于用户参与和用户主动地提供而不是通过大量转载和连接, 有少数的内容是由网站编辑人员来控制 and 发布的。

(3) 网站管理者对其内容很少进行强制性发布, 而是在平台搭建完成后, 对 UGC 的内容信息进行引导并促使其发生转变。

(4) 网站的推广和传播模式以用户间的相互传播为主, 在其产品使用过程中, 鼓励使用者进行用户群体之间的传播。

(5) 网站的主要盈利模式是基于以上 UGC 内容及信息大量聚合后所释放的能量而进行的, 无论是在线的广告业务、ISP 增值业务还是其他收费模式, 都需要在这个平台上发生。

(6) 注重用户体验的持续服务, 服务和应用无处不在, 不仅仅是少数重要用户, 而是渗透到全体用户, 包括大量的普通用户, 要有拉动长尾的能力。

Web 2.0 理念使得网站的展现形式更加多样化, 产生了很多的典型产品, 例如: 论坛、名人博客等。其实, 无论是 Web 1.0 还是 Web 2.0, 用户的体验是最重要的。随着互联网应用的更加普及, 人们不仅是信息的消费者, 越来越多的人会为网络提供信息, 成为信息的制造者和提供着, 并且因为参与了网络信息的提供而感到充实和自我实现。

### 1.4.2 语义 Web

万维网已经成为人类最大的信息仓库, 而且各种语言、各个知识领域的内容还在源源不断地快速增长。这些海量的信息在给我们提供便利的同时, 让我们的信息查询变得极为困难。大多数搜索引擎是基于关键词的搜索, 搜索是基于页面内容的, 而不是基于页面内容或页面信息的语义, 查准率较低。人们需要让页面内容有意义, 从而提供各种依靠语义的自动化服务, 这就是语义 Web 的研究动机。

#### 1. 语义 Web 的概念

在 WWW 出现不久, 人们就已经意识到语义对于 Web 的重要性。HTML 只是规范了信息的显示, 却无法表达内容的含义。没有形式化的网页内容, 机器将无法实现信息处理的自动化。只有将网页内容表述成机器可以理解的格式, Web 才可能成为一个巨大的知识库, 充分实现信息的查找、共享和重用。为此, 1998 年, 万维网的发明者 Tim Berners-Lee 首次提出了语义 Web (Web Semantic) 的概念。

对于语义 Web 的概念, 一般表述是: 语义 Web 是当前 Web 的一个扩展, 其中信息具有形式化定义的语义, 更有助于计算机之间以及计算机与人之间的协同工作。其思想是使 Web 上的数据以这样一种方式来定义与链接, 使其能够在各种不同的应用场景中有效地实现数据的发现、自动化处理、集成与复用。当且仅当 Web 不仅成为人所共享加工的场所, 也成为自动化工具所共享加工的场所时, 语义 Web 方能实现其全部潜力。

语义 Web 有很多突出的优点, 包括数据集成更简单、搜索更精确、知识管理更方便等,

结果语义 Web 这个词的含义越来越丰富。

## 2. 语义 Web 体系架构

要实现语义 Web，依赖于三大关键技术：XML、RDF 和 Ontology。语义 Web 分层体系架构如图 1-4 所示。

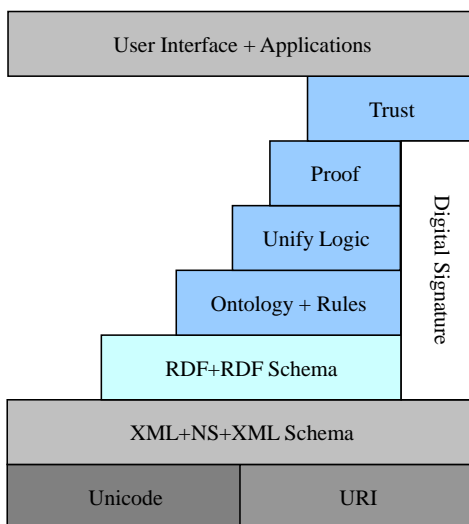


图 1-4 语义 Web 体系架构

(1) Unicode 和 URI 层。Web 内容采用 Unicode 字符集，负责资源的编码，统一资源定位符 URI (Uniform Resource Identifier) 用于资源标识，唯一标识网络上的一个概念或资源。在语义 Web 体系结构中，该层是整个语义 Web 的基础。

(2) XML+NS+XML Schema 层。该层可以称为 XML 层，用于表示数据的内容和结构。XML 实现文档的结构化定义，即进行文档格式化。命名空间 NS (Name Space)，由 URI 索引确定，目的是避免元素重名而引起的歧义。XML Schema 提供更多的数据类型，为 XML 文档提供数据校验机制。该层负责从语法上表示数据的内容和结构，实现 Web 内容和表现形式的分离。

(3) RDF+RDF Schema 层。又可以分为 RDF 层和 RDF Schema 层，其中 RDF 用于描述资源及其相互关系，RDF Schema 层为 RDF 提供类型定义机制，确定 RDF 描述的资源所使用的领域词汇。因为，XML 不具备语义描述能力，W3C 推荐以 RDF (Resource Description Framework) 标准来解决 XML 的语义局限。

RDF 解决的是如何采用 XML 标准语法无二义性地描述资源对象的问题，使得所描述的资源元数据信息成为机器可理解的信息。RDF Schema 使用一种机器可以理解的体系来定义描述资源的词汇，其目的是提供词汇嵌入的机制或框架，在该框架下多种词汇可以集成在一起实现对 Web 资源的描述。

(4) Ontology + Rules 层。本体 (Ontology) 负责在 RDF (S) 基础上定义的概念及其关系的抽象描述，用于描述应用领域的知识，描述各类资源及资源之间的关系，实现对词汇表的扩展。在这一层，用户不仅可以定义概念，而且可以定义概念之间的复杂关系。规则用于描述领域知识中的前提和结论。本体和规则共同构成领域知识层。

W3C 推荐使用 OWL (Web Ontology Language) 作为 Web 本体描述语言, OWL 既提供了正式的语义, 又提供了附加的词汇, 比起 XML、RDF 和 RDF Schema, 对 Web 内容实现了更好的机器互操作性。

(5) Unify Logic 层。该层负责在下面各层基础上, 提供公理和推理规则, 而 Logic 一旦建立, 便可以通过逻辑推理对资源、资源之间的关系以及推理结果进行验证, 证明其有效性。

(6) Proof 层。通过 Proof 交换以及数字签名, 建立一定的信任关系, 从而证明语义 Web 输出的可靠性以及其是否符合用户的要求。

(7) Trust 层。支持代理间通信的证据交换, 在用户间建立信任关系。

(8) User Interface + Applications。应用层是构建在语义 Web 之上的各种应用。

总之, 在语义 Web 体系架构中, 下面两层是语义 Web 的基础设施, 中间从元数据发展到本体描述语言及其统一的逻辑是语义 Web 的关键, 证明和信任及各层次贯穿的数字签名技术是扩充, 是对语义 Web 成功应用的要求与展望。

### 3. 语义 Web 的应用

在语义 Web 中, 可以提供各种依靠语义的自动化服务, 包括: ①互联网信息发布与搜索, 通过对内容的标注与分析从而克服关键词查询的歧义性, 可以大大提高查询精度。此外, 基于语义 Web 的文档检索与知识管理也是当前研究的一个热点。②Web 问题解答, 在用 Ontology 对信息源进行标注的基础上, 进一步运用知识库来解答用户的提问。例如, Stanford 大学研制的 Triple 系统是一个基于逻辑程序设计的 RDF 查询系统, 逻辑子句的问题求解能力使它能够解答较为复杂的问题。德国 Karlsruhe 大学等单位研制的 SEAL 是一个语义 Web 门户网站, 它具有回答用 F-逻辑表示的查询的能力, F-逻辑使得 Ontology 中的概念与问题求解规则融合于一体。

语义 Web 的目标是改善当今的 Web, 它的主要思想是使语义信息成为计算机可处理的对象。要将 Web 语义化是非常困难的, 语义 Web 很难一下子获得巨大成功, 但是, 它会一点点地渗透到现有的 Web 中, 在人们的不知不觉中, 让我们进入语义 Web 的时代。



本章从计算机技术发展的角度, 介绍了软件设计与开发模式的演化过程。对传统的单机程序和 C/S 模式下的程序设计进行了简单说明, 讲解了 Web 中 B/S 三层体系结构的工作原理及 Web 应用 (程序) 的基本结构, 为 Web 编程给出了一个主体的知识框架, 它是整本书写作的主线。介绍了 Web 中主要的关键技术, 包括: Java 技术及其主要特征、XML 技术及标准体系和 Web 服务, 以及 Web 的发展趋势, 包括 Web 2.0、语义 Web 等, 从而为后续章节的学习建立起一个整体知识框架和做好概念上的铺垫。



#### 一、填空题

1. 在传统的程序开发中, 一般要经历一个\_\_\_\_\_、\_\_\_\_\_和\_\_\_\_\_的开发过程。



2. 互联网的发展可以分为三个阶段, 分别是\_\_\_\_\_、\_\_\_\_\_和\_\_\_\_\_。
3. 在 Web 中, B/S 三层结构中的三层是指\_\_\_\_\_层、\_\_\_\_\_层和\_\_\_\_\_层。
4. 从本质上讲, Web 浏览器的功能可以分为两个方面, 分别是\_\_\_\_\_和\_\_\_\_\_。
5. Web 服务器程序是指\_\_\_\_\_。
6. 从技术上讲, 一个 Web 站点是\_\_\_\_\_构成的。
7. URL 的一般形式为\_\_\_\_\_。
8. Sun 的 JDK 可分为\_\_\_\_\_和\_\_\_\_\_两个部分。Java 进入了 Java 2 时代是以\_\_\_\_\_为标志的。
9. \_\_\_\_\_实现了 Java 的跨平台特性, 实现了 Java 程序的“编写一次, 到处运行”特性。
10. 在互联网的发展中, \_\_\_\_\_实现了代码的平台无关性, \_\_\_\_\_实现了数据的平台无关性。
11. XML 技术和 HTML 的本质区别是\_\_\_\_\_。
12. 在 XML 技术标准体系中, 按照标准的功能划分, 可分为\_\_\_\_\_、\_\_\_\_\_和\_\_\_\_\_三个层次。
13. 和 Web 1.0 相比, 在 Web 2.0 中, 用户不仅是网站内容的\_\_\_\_\_, 更是网站内容的\_\_\_\_\_。
14. 在语义 Web 中, 要实现信息的形式化定义, 其目的是\_\_\_\_\_。
15. 总体上讲, World Wide Web 是由互联网中成千上万的\_\_\_\_\_和\_\_\_\_\_组成的。

## 二、简答题

1. 简述软件设计与开发模式的演化过程。
2. 什么是 Web 服务器? 简述 Web 的基本工作机理。
3. 解释下列概念:  
网站、网页、HTTP、URL、端口、超文本
4. 什么是 B/S 结构? 和 C/S 结构相比, 有什么优点?
5. 什么是 Web 应用? Web 应用和传统的可执行程序有何不同?
6. 简述你对 Java 技术的理解。
7. XML 技术标准体系是如何划分的? 列举常用的 XML 基础标准, 并说明它们的功能。
8. 什么是 Web 服务? Web 服务基于哪些主要的技术标准?
9. 什么是 Web 2.0? 它有哪些主要特征?
10. 什么是语义 Web? 画出语义 Web 的分层模型, 并说明各层的功能。