



一线SAP顾问倾力写作
深度剖析SAP Java基础配置及开发技术

孙东文 郭娟 等编著

SAP

Java开发技术详解——Web Dynpro基础应用

SAP Java Development Guide: Web Dynpro Basic Application



SAP Java 开发技术详解

——Web Dynpro 基础应用

孙东文 郭娟 郭欢 张岩 编著



机械工业出版社

本书主要介绍了 SAP 异于 ABAP 传统编程的 SAP Java 的 Web 编程技术。围绕常用的 Web Dynpro for Java 技术,以 SAP Java 的技术架构、开发环境、基础编程和相关应用为主线,讲解 Web Dynpro for Java 这门技术中各元素之间的关联及制约关系,包括如何设置 Web Dynpro for Java 的运行环境,如何创建及编辑 Web Dynpro for Java 的基本元素,如何应用相关技术等。

技术架构部分由浅入深,详述了 Web Dynpro for Java 的相关概念,Web Dynpro 组件中视图、窗体、Web Dynpro 控制器相关组件的作用和关系。开发环境部分详细地介绍了操作系统和服务器的相关配置,并详述如何检查配置结果和配置效果。基础编程部分系统、全面地介绍了各 Web Dynpro 元素的原理及实现步骤,举例详述如何创建各 Web Dynpro 元素,并详述不同 Web Dynpro 元素的编辑实现及应用效果。基础应用部分介绍了 Web Dynpro 组件之间进行数据交换的几种方式,对 Web Dynpro 组件、Web Dynpro 组件接口及无接口的组件的使用办法、实现方式进行了详细阐述,并举例说明每种方式的编辑实现及应用效果。

本书提供了大量配置细节及实例源代码,深入剖析 Web Dynpro 架构关键技术,技术架构部分为 SAP Java 开发人员深入 SAP 系统的必备指南。适用于 Java 开发人员转型为 SAP 以及 EP 和 CRM 平台下 Web Dynpro 和 Web UI 程序的开发人员。

为了更好地帮助读者学习,本书对 WDJ 所涉及的 Basis 相关技术做了较多的阐述。更多相关 Basis 技术知识还需参阅相关资料。

图书在版编目(CIP)数据

SAP Java 开发技术详解: Web Dynpro 基础应用 / 孙东文等编著. —北京: 机械工业出版社, 2019.3

ISBN 978-7-111-62777-7

I. ①S… II. ①孙… III. ①JAVA 语言—程序设计 IV. ①TP312.8

中国版本图书馆 CIP 数据核字(2019)第 096030 号

机械工业出版社(北京市百万庄大街 22 号 邮政编码 100037)

策划编辑: 车 忱 责任编辑: 车 忱 李晓波

责任校对: 张艳霞 责任印制: 郜 敏

北京圣夫亚美印刷有限公司印刷

2019 年 7 月第 1 版·第 1 次印刷

184mm×260mm·22.5 印张·557 千字

0001—1000 册

标准书号: ISBN 978-7-111-62777-7

定价: 119.00 元

电话服务

客服电话: 010-88361066

010-88379833

010-68326294

封底无防伪标均为盗版

网络服务

机 工 官 网: www.cmpbook.com

机 工 官 博: weibo.com/cmp1952

金 书 网: www.golden-book.com

机工教育服务网: www.cmpedu.com

前 言

随着 SAP 软件在国内企业的广泛应用，SAP 实施及运维的 IT 服务行业异军突起，SAP 相关技术也得到了广泛应用，其中不乏 SAP Web 开发的高端技术。本书就是在这种情况下诞生的。

作者根据 SAP 官方教材、SAP 项目开发经验和 SAP 技术的升级，对 SAP Java 开发技术做了分类，将 Web Dynpro 这一单独的 SAP Web 编程技术纳入本书。

以往 Java 的书籍仅对 Web 编程简单做一下概念上的剖析，而对 SAP Java 的原理、架构及实现的介绍少之又少。本书特点在于对其技术原理阐述、实施前提、实施步骤、实施结果作了详尽的说明。更有作为 SAP Web Dynpro Java (SAP WDJ) 开发约定俗成的命名规则，SAP WDJ 的一般开发步骤及 SAP WDJ 开发常用的 UI 组件说明，以帮助那些做技术提升的 SAP 开发人员。

学习 SAP Web 开发技术，需要开发者自身水平的提高（例如：从面向过程的编程到面向对象编程的转变；从面向对象的语言开发到基于 MVC 的理论的 Web 开发之间的转变及 MVC 理论下的数据关系映射的了解），也需要熟悉 SAP 相关操作（例如窗体和视图的创建和调试），更需要 SAP 开发者对各种技术实施后最终体现结果的认同（例如如何应用创建的导航链接；Web 开发技术实施应用以后有什么样的效果，是否和需求吻合）。

本书中很多技术名词在以往的文献中鲜有记录，有的术语也是首次翻译成汉语，为了便于读者理解，易于应用，很多术语在第一次出现时冠以汉语+英文；有的为体现其本意如 Context 直接用的是英文，而没有用通常的中文译词“上下文”。

参与本书编写的有孙东文、郭娟、郭欢、张岩，由于作者水平有限，书中不免有疏漏和不足，敬请读者斧正。

编 者

目 录

前言	
第 1 章 Web Dynpro for Java 的由来	1
1.1 Java	1
1.2 NetWeaver	2
1.3 Web Dynpro	2
1.3.1 Web Dynpro 的双重目的	3
1.3.2 Web Dynpro 与其他 Web 开发工具的区别	4
1.3.3 Web Dynpro 应用程序的开发周期	4
第 2 章 SAP Web 应用服务器	7
2.1 概览	7
2.2 架构	7
2.2.1 SAP Web 应用服务器	7
2.2.2 SAP Web 应用服务	8
2.3 SAP Web 应用服务器的演变	8
2.3.1 系统演变	8
2.3.2 组件演变	9
2.3.3 集成环境	12
2.4 SAP Web 服务器架构和管理工具	13
2.4.1 技术系统架构	14
2.4.2 SAP Web 应用服务器管理工具	14
第 3 章 SAP Java 开发环境	19
3.1 开发平台	19
3.2 基于 Eclipse 平台的体系结构	20
3.3 插件	21
3.4 Eclipse 用户界面	22
3.5 SAP NetWeaver 开发工作室	23
3.5.1 基本概念	24
3.5.2 工具集	25
3.5.3 项目管理	28
3.5.4 Web Dynpro 开发工具	28
3.6 Web Dynpro for Java 项目的两种基本类型	33
3.6.1 本地 Web Dynpro 项目	33
3.6.2 基于组件的 Web Dynpro 项目	34
第 4 章 Web Dynpro 基本概念	37

4.1	Web Dynpro 架构	37
4.1.1	MetaModel 的概念	38
4.1.2	平台独立性	38
4.1.3	Web Dynpro 客户端	38
4.1.4	图形化开发工具	38
4.1.5	业务逻辑与应用逻辑分离	39
4.1.6	MVC 模型的转换	39
4.2	Web Dynpro 组件	41
4.2.1	Web Dynpro 组件特性	42
4.2.2	视图	42
4.2.3	窗体	44
4.2.4	Web Dynpro 控制器	47
4.3	Web Dynpro 组件接口	54
4.4	Web Dynpro 应用程序	56
4.5	Web Dynpro 建模	57
4.6	Web Dynpro 建模工具	57
第 5 章	导航	59
5.1	导航建模工具	59
5.1.1	打开导航建模工具	59
5.1.2	插头	60
5.1.3	入站插头、事件和动作	60
5.2	开发实例	61
5.2.1	开发要点	61
5.2.2	实例开发	62
第 6 章	Context	78
6.1	Context 简介	78
6.1.1	控制器 Context 编辑器	78
6.1.2	打开 Context 编辑器	79
6.1.3	属性类型	81
6.1.4	节点属性	82
6.1.5	节点与属性接口	84
6.2	数据绑定	86
6.3	Context 映射	87
6.4	实例	89
6.4.1	开发要点	89
6.4.2	实例开发	89
第 7 章	控制器	127
7.1	模型-视图-控制器	127
7.2	Web Dynpro 应用程序	128
7.3	控制器的类型及结构	130

7.3.1	视图控制器	133
7.3.2	接口控制器	134
7.4	控制器的方法和属性	135
7.4.1	控制器的默认方法	135
7.4.2	控制器成员变量	137
7.5	访问路径	139
7.6	实例	142
7.6.1	开发要点	142
7.6.2	实例开发	143
第 8 章	用户界面控制	174
8.1	UI 控件集	174
8.2	数据绑定	176
8.3	实例	179
8.3.1	开发要点	179
8.3.2	实例开发	179
第 9 章	用户界面模型	204
9.1	基本术语	204
9.2	视图集	207
9.3	实例	209
9.3.1	开发要点	210
9.3.2	实例开发	210
第 10 章	错误消息处理	240
10.1	消息编辑器	240
10.2	抛出消息的函数	241
10.3	实例	243
10.3.1	开发要点	243
10.3.2	实例开发	243
第 11 章	一般用户界面服务	254
11.1	简易静态值的数据集实现	256
11.2	扩展值动态值的数据集实现	257
11.3	实例	258
11.3.1	开发要点	258
11.3.2	实例开发	258
附录		268
附录 A	SAP NetWeaver 2004s (7.0) 安装配置指南	268
附录 B	IDE (SAP NetWeaver Developer Studio) 安装配置指南	304
附录 C	Web Dynpro for Java 命名规范	314
附录 D	创建第一个 Web Dynpro 程序	317
附录 E	用户界面元素	348

第 1 章 Web Dynpro for Java 的由来

Web Dynpro 是 SAP 传统 Dynpro 编程方法在互联网时代的扩展。SAP 为 Web Dynpro 提供两种开发语言：ABAP 和 Java。于是就有了 Web Dynpro for ABAP 和 Web Dynpro for Java。

从技术角度来讲，Web Dynpro for ABAP/Java 是 SAP 开发基于 Web 界面的一次革命。它完全不同于以前的开发模式，是 SAP 在 ERP 应用中以 Web 界面开发的一次飞跃。它适用于通过声明方式生成标准用户界面（User Interface，用户界面，简称 UI），以及缩短实现 Web 应用程序所需的时间。

Web Dynpro 基于 MVC 规范，使用了声明性的编程技术。当在网页上指定了要显示什么样的 UI 元素，在处理中设定好这些 UI 元素怎样取得数据，在应用中声明可能的浏览路径后，所有 UI 元素所需要的代码会自动生成，使程序员从重复的编码中解脱出来。可以指定客户端包括哪些 UI 元素，以及这些元素将从哪里获取数据。还可以通过声明的方式定义应用程序中可能的导航路径，然后，用于创建 UI 的所有代码将在标准运行时框架内自动生成，减少程序员在编写 HTML 代码时的重复编码工作量，并使其能够与 JavaScript 进行交互。

它与 SAP 以往的设计模式完全不同，代表了开发基于 Web 的 ERP 应用所取得的重大突破。基于 MVC 设计模式的 Web Dynpro for Java 是基于表单的 UI 开发的重大突破，具有革命性的意义。

注：SAP Java 开发的内容比较多，除了常见的 J2EE 相关的开发之外，还包括 SAP 特有的 Portal 开发、Web Mobile 开发等，本书围绕 Web Dynpro for Java 这门技术对 SAP Java 进行相应的阐述。

1.1 Java

Java 是一种可以编写跨平台应用程序的面向对象程序设计语言，是由 Sun Microsystems 公司于 1995 年 5 月推出的 Java 程序设计语言和 Java 平台（即 JavaSE、JavaEE、JavaME）的总称。Java 技术具有卓越的通用性、高效性、平台移植性和安全性，广泛应用于个人计算机、数据中心、游戏控制台、超级计算机、移动电话和互联网等，同时拥有全球最大的开发者专业社群。在全球云计算和移动互联网的产业环境下，Java 更具备了显著优势和广阔前景。

2007 年开始，SAP 的应用服务器平台已经转向了基于 J2EE 的 NetWeaver 的 SOA 平台，Java 也随即成为 SAP 中 EP（Enterprise Portal）及 KM（Knowledge Management）的开发语言，SAP Java 技术也因相关应用的普及而受到关注。

1.2 NetWeaver

NetWeaver 是 SAP 所有应用的技术集成平台的名称。SAP 的所有产品，包括 ECC（ERP Core Component，企业资源计划核心组件）、CRM（Customer Relationship Management，客户关系管理系统）、SRM（Supplier Relationship Management，供应商关系管理系统），以及所有的解决方案，包括 mySAP ERP、mySAP Business Suite 等，都基于这一平台。

SAP NetWeaver 是一种可随时用于业务运作、面向服务的平台，用于 SAP 的所有解决方案。SAP NetWeaver 平台内嵌了商务智能（BI）以及无线射频识别（Radio Frequency Identification, RFID）功能，能够有效地进行主数据管理。同时，不同业务角色的用户可以通过企业门户（Enterprise Portal, EP）上网访问企业内的各种信息。SAP NetWeaver 是近年来具有革命意义的基础应用和集成平台产品，它建立了新的面向服务的 SAP 企业服务信息系统基础架构（Enterprise Service Architecture, ESA），提供了一种完全开放而且灵活的基础设施，加强了对各层面的 IT 标准和行业标准的支持，使各公司能够通过现有的 IT 投资获取附加值，从而降低了企业的 IT 总体拥有成本。它是个应用平台，而且 DB（Database）具有 OS 无关性，通过与 .NET 和 J2EE 的互动增加了灵活性；同时它又实现了人员集成（多渠道访问、企业门户、协作等）、信息集成（商业智能、知识管理、主数据管理等）和流程集成（集成代理和商业流程管理等）。

Web Dynpro 就是基于这一平台，在 SAP 的各种解决方案中应运而生的。

1.3 Web Dynpro

SAP Java 技术就是 SAP NetWeaver 平台下的 Java 开发。SAP Java 开发遵循 Java 的标准，可以在 SAP NetWeaver 平台上开发，部署企业级 Java 应用——J2EE。

Web Dynpro for Java 是在 SAP NetWeaver 平台下 Java 开发中使用最多的技术。

Web Dynpro 开发 UI 是基于 MVC 的 SAP 标准用户界面技术。Web Dynpro for Java 应用程序使用模型驱动的方法，使用可视化工具最大限度地减少手工编码。Web Dynpro 运行于 SAP NetWeaver 平台，是 SAP 用来开发大型企业应用的技术成果，简化了 SAP 传统技术的开发程序，从而降低了总成本，提高了生产效率。Web Dynpro 目前在市场上已被越来越多的企业所采用。

Dynpro 通俗地讲就是屏幕主界面。一般的系统窗体包括标题栏、菜单栏、工具栏、状态栏和主界面。Dynpro 特指 SAP GUI 主界面区域的内容，是 ABAP 程序的一种。

Dynpro 是由 UI 元素和其背后的商业逻辑组成的动态程序。Dynpro 的主要组成部分包括：

- 1) 属性。例如当前的屏幕编号，以及下一个屏幕的编号等。
- 2) UI 元素的位置。例如文本标签、文本输入框、按钮等界面元素的位置安排。
- 3) UI 元素的属性。
- 4) 程序逻辑。这些程序将调用后台的 ABAP 代码模块。

而 Web Dynpro 采用了一种高级的 MVC/Data Binding 架构模式，并且提供非常友好的编程界面。Web 界面可以使用拖拉的形式进行开发。由于 Web Dynpro 运行于 SAP NetWeaver 平台，该平台在传统 Dynpro 开发上的优势也被发挥得淋漓尽致。使用 Web Dynpro 可以非常迅速地开发出企业级的应用程序，开发者只需要关注其业务流程，而版本管理、质量控制、发布、性能等内容由 NetWeaver 平台帮助完成。

Web Dynpro 支持结构化开发。Web Dynpro Component 通过模块化，被组合起来形成复杂的 Web 应用。

Web Dynpro 是以声明的方式进行开发的，ABAP 开发平台提供了一些工具并以其独特的方式来抽象地表示相应的应用。所有需要的代码会自动创建并符合 Web Dynpro 框架的标准。

Web Dynpro 框架允许程序员将自己的源代码放在自动生成的代码的指定位置。所有的 Web Dynpro 应用都基于相同的逻辑单元，然后通过加入自己的编码，程序员就可以扩展这个框架来满足自己的业务需求。不是在设计的时候就要决定所有的实施细节，可以在运行的时候动态决定界面的显示。这样就使程序的应用变得非常灵活而不需要写 HTML 或 JavaScript 代码。

Web Dynpro 可以访问不同的数据源。

所有可重用的部件都可以被调用（如 Function 或 Method），甚至通过 Open SQL/SQLJ（SQLJ 是一个与 Java 编程语言紧密集成的嵌入式 SQL 的版本）直接从数据库中取得数据。然而这样会导致程序逻辑和业务逻辑混淆，所以应当避免。

通过创建 Web Service Client Object 可以访问 Web Service。

SAP Java Connector (JCo) 可以调用 Java Engine 上的 Enterprise JavaBeans。Model Object 在 Web Dynpro for ABAP 中还不被支持。最好的方式是通过封装业务逻辑的可重用的实体来创建包含源代码的类。也可以开发无窗体的 Web Dynpro Component，其目的只是为了重用。这些 Component 可以以 Component Usage 的方式被访问。

注：通过路径 Window->perspective 更改开发环境的外观，适合于不同的开发项目类型；通过路径 Window->Show View 显示相关开发工具视图。

1.3.1 Web Dynpro 的双重目的

Web Dynpro 的主要目的是在结构化设计的方式下使开发人员以最少的代码能够开发功能强大的 Web 应用，让开发人员能够在结构化设计流程中使用声明式工具，从而减少工作量。Web Dynpro 理念的一个指导原则是：手写代码行数越少，程序功能越强。Web Dynpro 以两种方式实现这一目标：

1) Web Dynpro 使用声明式的、语言中立的元模块来定义 UI。开发环境根据这一抽象定义生成所需的源代码。手写代码仍然有一席之地，但是只限于操作业务数据，而不适用于 UI。

2) Web Dynpro 提供诸多技术功能，例如支持国际化、无闪烁交互，以及明确分离业务逻辑和 UI 等。该分离可通过对模型视图控制器 (MVC) 设计范例的实施加以修改来实现。

Web Dynpro 的另一个目的便是尽量减少手工敲入代码。Web Dynpro 通过下面两种方式

来实现这一目标：

1) 尽可能避免对 UI 层进行编码，为了开发 UI，Web Dynpro 提供了一个公共的元数据模型，这样就直接导致几乎无须为 UI 编写程序代码。

2) 允许业务应用程序以独立于后台业务平台的前端表示层的形式存在，由于已经消除了编写 UI 代码的重复劳动，开发者可以将精力集中于应用程序的业务数据流。Web Dynpro 应用程序可以在各种设备、各类网络上运行——这是协作应用的重要功能。

由于 Java 技术的广泛性和跨平台性，使得 Web Dynpro for Java 技术的应用更加广泛，SAP 的 Web Dynpro for Java 开发的应用程序通过企业门户 (EP) 可以独立在浏览器中被访问。这样，企业特别是国内的大型企业，业务流程复杂多变，通过使用 Web Dynpro for Java 技术开发符合自己特定业务流程的应用，并且不需要安装 SAP 客户端就可以轻松访问，大大地提高企业的工作效率，也降低企业的运行成本。

1.3.2 Web Dynpro 与其他 Web 开发工具的区别

从开发者的角度来看，在其他开发工具中 (如 JSP)，是以 Web 页作为开发单位，而且用户的应用程序由一套已经被链接的页面组成，这些页面共同提供所需要的业务功能。

然而，在 Web Dynpro 中，是以组件 Component 为开发单位的，这里的 Component 是指一套相关的 Java 程序，这些程序一起形成可重用的业务功能。一个 Component 可以没有或者具有多个视图，从这一点上，Web Dynpro Component 可以认为是相关 Web 页的聚合。Web Dynpro 框架允许程序员将定制源代码置于已生成的代码内的预定义位置。所有 Web Dynpro 应用程序都使用相同的基础单元构建，然而，使用定制代码可以扩展标准框架以提供自己所需的业务功能。并非所有实施决策都要在设计时制定，可以先实施 Web Dynpro 应用程序，用户界面的外观可以在运行的时候决定。这样，不必直接编写任何 HTML 或 JavaScript 代码，即可编写高度灵活的应用程序。

1.3.3 Web Dynpro 应用程序的开发周期

SAP 为开发的所有过程和生命周期提供了一个集成和强健的框架，一个开发周期通常包括：分析阶段，设计阶段 (包括架构设计，详细设计) 和实施阶段。

1. 分析阶段

在分析阶段，应确定并描述应用程序的业务需求。在这一阶段，仅需考虑是何种业务流程，而不要考虑如何设计此流程。

此阶段的成果为分析模型，使用非技术人员就可以理解的语言完整、准确、统一地描述业务需求。

由于分析模型不考虑业务流程是怎样的，所以此阶段进行的描述与任何特定技术 (比如 Web Dynpro) 无关。

2. 设计阶段

(1) 架构设计阶段要解决的问题

1) 为 Web Dynpro 开发组件 (DCs) 建模。

- 需要哪些开发组件来交付所需功能？
- 是否需要开发新的组件？是否可以重用现有开发组件？

- 要使用开发组件的哪些层次结构排列？
- 各开发组件之间存在哪些相关性？
- 哪些公共部分已发布？

2) 为 Web Dynpro 项目建模。

- Web Dynpro 项目由哪些已经使用的 Web Dynpro 组件组成？
- 哪些 Web Dynpro 组件包含其他 Web Dynpro 组件的接口视图？
- Web Dynpro 项目中使用哪些模型？
- 哪些 Web Dynpro 组件使用哪一模型？
- Web Dynpro 项目中定义哪些 Web Dynpro 组件接口？
- 哪些 Web Dynpro 组件使用哪些 Web Dynpro 组件接口？
- 各个 Web Dynpro 组件之间存在哪些调用方法？各个 Web Dynpro 组件之间使用哪些结果？
- Web Dynpro 组件之间存在哪些 Context 映射？

(2) 详细设计阶段要解决的问题

1) 为 Web Dynpro 开发组件 (DCs) 建模。

- Web Dynpro 组件由哪些视图组成？
- Web Dynpro 组件中使用哪些自定义控制器？
- Web Dynpro 组件内需要定义哪些方法和事件？
- 在 Web Dynpro 组件内的哪些控制器中定义 Context 元素？
- 在 Web Dynpro 组件内的哪些组件接口中定义 Context 元素？
- 绑定或者映射哪些 Context 元素？

2) 为 Web Dynpro 窗体建模。

- 应如何安排视图？
- 各视图之间存在哪些导航路径？

3) 为 Web Dynpro 视图建模。

- 视图中包含哪些用户界面元素？
- 应如何安排视图的用户界面元素？
- 用户界面元素与哪些 Context 绑定？
- 用户界面元素绑定到哪些 Action 中？

一旦解决好上述问题，就可以使用 SAP NetWeaver Developer Studio 中的图形窗体和组件建模工具开始构建应用程序了。

3. 实施阶段

(1) 用户界面 (UI) 元素 (Element): UI 元素应使用标准命名规约

应该在创建 UI 之前创建 Context 元素，可以使用 Web Dynpro 向导生成 UI 元素，从而简化 UI 的开发；应用容器元素属性以控制 UI 子元素的布局，从而避免不必要的改动与维护；所有按钮和主要的交互元素都应包含图像，以提高直观性。

用于弹出消息的视图应该是通用的。这些视图中的所有 UI 元素都应绑定到 Context 节点以便重用。

(2) 数据: Context 节点和属性应使用标准命名规约

应通过在 Context 内存中保存较大的数据集来避免多次访问数据库或其他外部对象以提高效率；通过使用 Context 内存而不是通过 ABAP 内表缓存来存储数据；通过供应函数填充内存数据，而不是在初始化方法中嵌入抽取数据的代码。

在组件控制器中创建标准 Context 节点并映射到视图。通常要创建 2 个 Context 节点，一个 Context 节点用来保存与视图属性相关的数据，另一个用于保存用户选择的信息。

简单的数据类型可以存储在 Web Dynpro 属性中，而不是存储在 Context 中，这减少了访问数据所需的代码量。

(3) 方法 (Methods) 和动作 (Actions): 应使用标准命名规约

Web Dynpro 应用程序与其他外部对象进行数据的交互应封装在类的对象中，例如数据库访问、程序执行等；使用事件触发其他 Web Dynpro 视图中的私有方法，而不是在组件控制器中创建共有方法。

更新或选择 Context 元素的公共调用可以封装在一个方法中，从而减少所需的代码量；导航应在视图的 WDDOBEFOREACTION 方法中执行验证。如果验证失败，则应调用取消导航。

UI 元素的输入验证应链接到 UI 元素，以突出显示错误的用户输入。使用文本符号而不是用消息类来保存提供语言翻译的文本。

注：可使用 SAP NetWeaver Developer Studio 导入、生成、部署和运行 Web Dynpro Application。关于 Web Dynpro for Java 的命名规范，可参阅附录 C。

第 2 章 SAP Web 应用服务器

2.1 概览

SAP Web 应用服务器的基本功能是基于 SAP 标准解决方案编程语言（通常是 ABAP）的 Web 开发，但新版本也允许使用 Java 语言进行开发。有了 Web 应用服务器，在 Web 环境下基于 SAP 解决方案基础设施的程序发布和 ABAP 应用就有了可能。新版的 WAS（Web Application System）也支持像 Java 这样的开放标准。6.20 以上的版本里可以单独创建 ABAP 程序、Java 程序或者两者的混合，7.0 以后统称为 NetWeaver。

2.2 架构

SAP Web 应用服务器不是 Web Dynpro 项目简单的开发环境。它提供了复杂的系统和友好的用户交互方式。

程序员依靠通用的工具和特性，如网页设计，高度可扩展的基础架构，在开放、集成、安全和独立的平台上对项目进行实现。

2.2.1 SAP Web 应用服务器

SAP Web 应用服务器的架构可以分为 5 层，如图 2-1 所示。

(1) 表示层

在表示层，用户接口可以使用 JSP，BSP 或者 Web Dynpro 技术来开发。下面的业务层通过 Java 或 ABAP 程序来提供业务内容。

(2) 业务层

业务层包括一个通过 J2EE 认证的运行时环境，该环境接受从 ICM 传来的请求然后动态产生一个应答。业务逻辑可以用 ABAP 或者基于 J2EE 标准的 Java 来实现。开发人员实现了业务逻辑后可以用 J2EE 环境的 EJB 来持续改进这些业务逻辑。开发人员还可以访问运行在 ABAP 环境里的应用的业务对象以便从它们的业务逻辑和持续改进中获益。

(3) 集成层

本地的集成引擎是 SAP Web 应用服务器的一个完整组成部分，可以实现和 SAP XI（Exchange Infrastructure，交换架构，是一款 SAP 的中间件产品，简称 XI，在 2005 年 10 月更名为 PI）的即时连接。本地的集成引擎提供信息服务，可以实现连接到 SAP XI 上各组件之间的信息交换。

(4) 链接层

因特网通信管理器（Internet Communication Manager，ICM）把用户接口的请求发送到

表示层并提供一个单一的框架来处理使用不同通信协议的各种链接。现在，可用的模块有 HTTP、HTTPS、SMTP、SOAP 和 FastCGI。

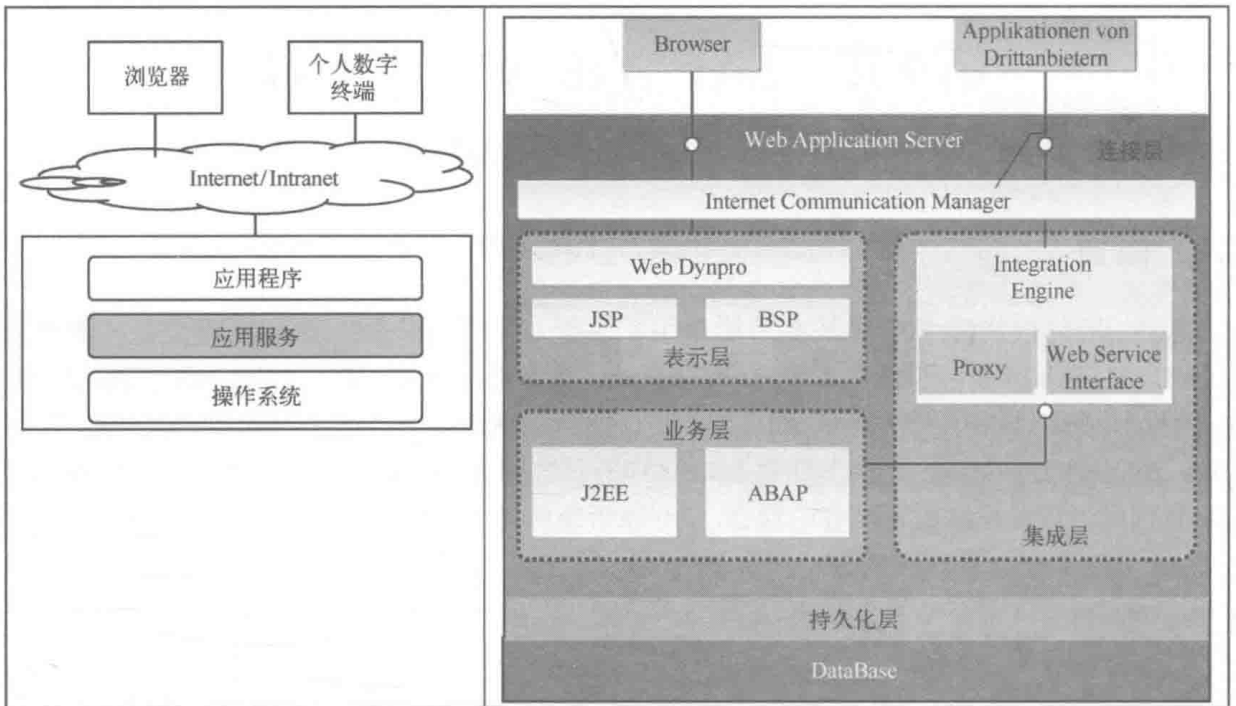


图 2-1

(5) 持久化层

持久化层支持数据库无关操作和弹性事务处理。业务逻辑的开发可以完全不考虑底层数据库和操作系统，对开放标准的支持也使数据库无关成为可能。数据库接口保证从 ABAP 环境里通过 Open SQL 进行的数据访问都是经过优化的。SAP 还推出了 Open SQL 对 Java 的支持，为应用开发人员提供了多种标准 API，如 SQLJ。其他技术，比如 JDO 和 CMP EJB，或者直接使用 JDBC API 访问都是支持的。

2.2.2 SAP Web 应用服务

SAP Web 应用服务有以下特性：

- 1) 支持新的应用开发、集成、电子商务和 B2B 集成。
- 2) 具有可靠性、可扩展性、可用性、安全性和应用适应性。

2.3 SAP Web 应用服务器的演变

2.3.1 系统演变

(1) 1972 年~1989 年

R/2 时代，实时数据处理 (Real Time Data Processing)，平台为 IBM 大型机，基于字符的用户界面，应用层以模块构成，开发语言为 ABAP，运行时环境是 Basis/ABAP。

(2) 1990 年~1998 年

R/3 时代，平台为三层客户/服务器系统，图形用户界面（GUI），应用层以模块构成，开发语言为 ABAP，运行时环境是 Basis/ABAP，对外接口为 ALE/RFC 与 IDoc。

(3) 1999 年~2002 年

mySAP.com 时代，平台为多层客户/服务器系统，图形用户界面及浏览器，应用层以模块及组件构成，开发语言为 ABAP/4，运行时环境是 Basis/ABAP，对外接口为 RFC/BAPI。

(4) 2003 年~现在

mySAP Business Suite/SAP NetWeaver 时代，平台为多层客户/服务器系统，用户界面是企业门户、图形用户界面及 Web 浏览器，应用层以组件构成，开发语言为对象化的 ABAP 及 Java，运行时环境是 WebAS/ABAP/J2EE，对外接口为 Web Services。

如果说 1999 年 mySAP.com 的技术革新，是为了应对网络商务时代的来临，那么 2003 年开始全面推出的 SAP NetWeaver 以及它所支持的企业服务架构（ESA），是 SAP 领导业界完成对面向服务架构（SOA）和 Web 服务的转变，同时也为了完成从 ERP 至 ERP II 的转型。Gartner 对 ERP II 的定义是传统 ERP 的组件化与公开化。将新的 mySAP 商务套件 + SAP NetWeaver 结构与原先的 R/3 + Basis 结构并列，方便分析比较，如图 2-2 所示。

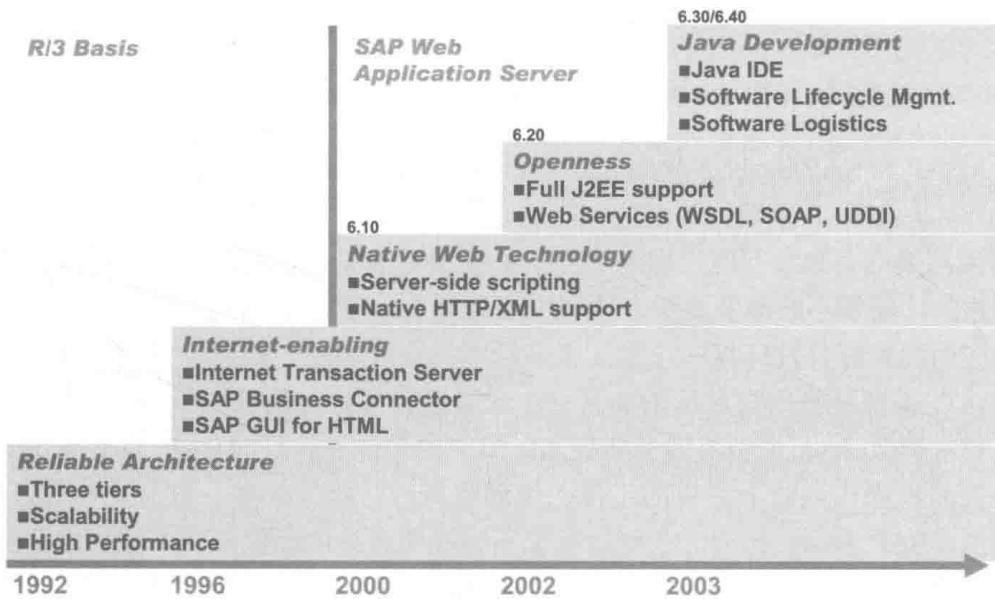


图 2-2

2.3.2 组件演变

SAP NetWeaver 不仅仅是 Basis 的简单延伸，其最底层的应用平台（Web AS）实现了对传统 Basis 功能的延展并加强，SAP NetWeaver 的其余三层集成结构：人员集成、信息集成和流程集成，大大丰富了 Basis 的功能。

从技术的角度简单介绍 SAP NetWeaver 一些组件的演变历程。

(1) SAP R/3 Open SQL 成为 SAP NetWeaver Open SQL

SAP R/3 Open SQL 和 SAP NetWeaver Open SQL 具有类似的功能，Open SQL 好像一个

数据和数据类型的字典，提供一个数据读写的抽象模块。因为使用 Open SQL，开发者可以写一个应用程序，在不同的数据库上运行。SAP NetWeaver Open SQL 同时支持 ABAP 和 Java 两种程序语言，如图 2-3 所示。

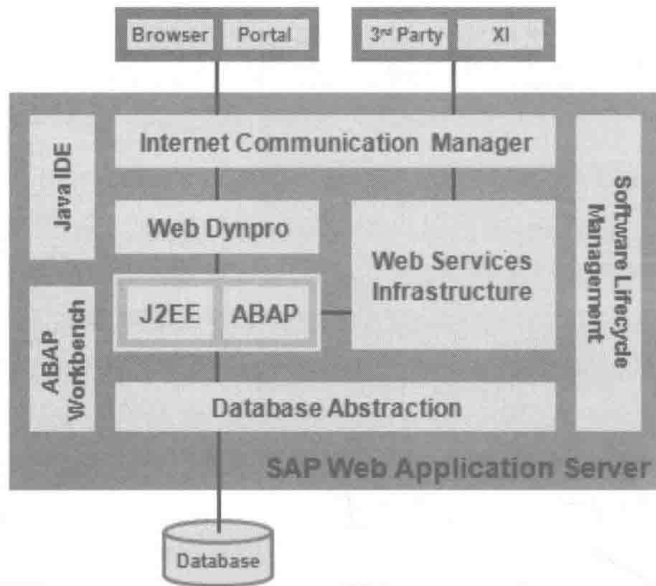


图 2-3

(2) ABAP 和 SAP Basis 演化为 SAP Web 应用服务器和 Java

ABAP/4 是由 SAP 发明的第四代程序语言，开发者可使用它编写商业应用程序。最初，SAP 公司用汇编语言来编写应用程序和用宏汇编编写抽象化模块。ABAP 是基于 Cobol 的编程思想创建的，随着不断地发展更多地利用了类似 Java 语言面向对象的特征。ABAP 语言是面向商务应用的友好开发环境，它包括很多标准的商务功能，如货币转换、日历、国际化特征等。ABAP 很容易将用户对话附加到一个应用逻辑模块。Basis 是一个操作系统的抽象化模块，它非常类似 Java 虚拟机的早期版本，但它的设计是运行 ABAP 而不是 Java。所有的应用程序需要在操作系统上完成的工作，像创建流程、收发邮件、打开文件等，SAP Basis 都能够提供。SAP Basis 作为一个可运行于不同操作系统的抽象模块，被标准化为一个应用服务器。SAP 新的应用服务器——SAP Web 应用服务器，是一个基于 J2EE 标准和 ABAP 的服务器。这个服务器应用自己的虚拟机和一系列开放标准提供操作系统功能，如网络浏览、过程控制等。SAP Web 应用服务器是 NetWeaver 最底层架构，专门负责与操作系统和数据库通信。SAP Web 应用服务器与 Basis 有一个包装及行销上的差别：Basis 是与 R/3 包装在一起，但 SAP Web 应用服务器可以单独销售，因为它本身是一个符合 J2EE 标准的应用服务器，含有基于 Eclipse IDE 标准的 NetWeaver Developer Studio，可与 IBM WebSphere，BEA WebLogic，Jboss，Apache Tomcat 等应用服务器产品分庭抗礼。它是很多基于 J2EE 标准的应用服务器中的一个，SAP Web 应用服务器与其他应用服务器最大的差别是它同时支持 Java 和 ABAP。

(3) 远端功能调用演化为 Web 服务

远端功能调用 (RFC)，是指一个企业应用允许另外的应用调用其功能，开发者可以利