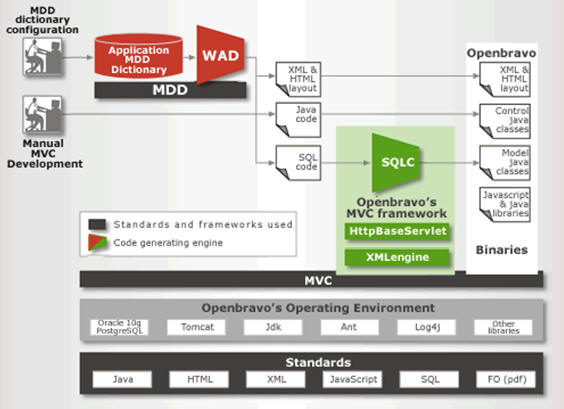
# ****概述****

Openbravo ERP是一套适合于中小企业并且基于web可扩展的开源ERP系统。这个ERP系统所包括的强大功能可实现生产管理、仓库管理、销售管理、财务管理。同时内置CRM(客户关系管理)和BI(商业智能)。在2008年获得了1200万美金的风险投资，并在2009获得了Infoworld的最佳开源企业软件的奖项，其前景相当看好。

研究同行业的开源软件，有助于我们掌握其产品架构，了解其设计理念，从而积累软件设计经验，以应用到自己的产品开发中。



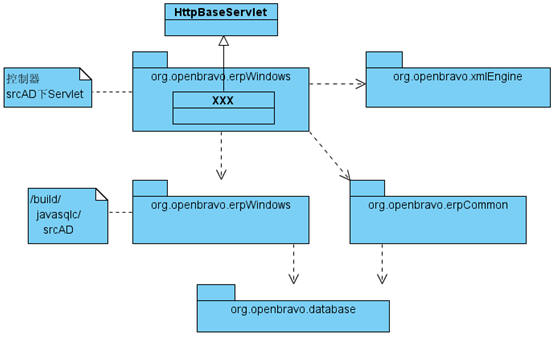
**图 1.1 Openbravo的体系结构**

Openbravo主要的体系结构被称为“应用数据字典”（Application Dictionary，简称AD）结构。这种结构源于另一个开源ERP Compiere ERP（值得一提的是，Compiere ERP也在09年获得了Infoworld的最佳开源企业软件奖）。**定义的数据字典将通过Openbravo的WAD引擎生成各种元素。开发者开发的SQL语句将通过SQLC程序生成对应的各个业务组件。所有的程序组装在一起就组成了Openbravo ERP。**

我们将从**ERP系统结构**和**编译系统结构**两部分来分析Openbravo的软件架构。

# ****ERP系统结构****

## 概述



**图 2.1 Openbravo运行时的逻辑视图**

系统运行时主要分为3层：

* **表现层：**主要由**org.openbravo.erpWindows**包下的控制器组成，他们继承自HttpBaseServlet。负责使用请求初始化数据，调用业务层，最终使用xmlEngine渲染显示页面。
* **业务层：**负责提供业务中的数据结构，如资产等。Openbravo的业务层和SQL耦合较为紧密，通常会在业务对象类的静态方法中直接执行SQL。处理业务也使用了存储过程，业务对象类里边也会有响应的静态方法执行存储过程（通常会根据数据库类型进行判断）。业务对象可以由编译引擎生成。
* **基础层：**基础层提供一些基础的服务，如提供数据库连接等。

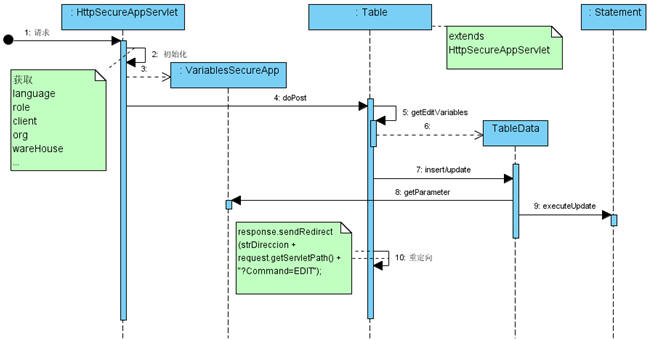
## 运行时处理过程

以在OpenbravoERP中**编辑数据字典时的程序处理**为例，运行时处理过程大概如下：

1. 首先请求被发送**每个程序对应的Servlet**，如Table，Table继承自一个模板类HttpSecureAppServlet，依靠这个类执行一些必要的初始化工作。响应请求的主要方法为doPost，doPost方法根据名称为Command的参数执行不同的操作。如SAVE\_NEW\_EDIT表示首先保存一条新增的记录，然后跳转至EDIT页面。

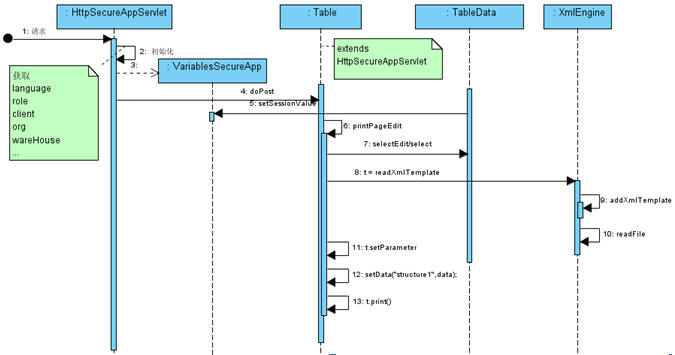
对于一个典型的保存请求（即Command参数的值以SAVE开头）：

如图 2.2，在doPost方法中首先使用getEditVariable方法从请求中构造业务数据对象TableData，然后调用其insert/update方法，这个方法将会调用其内部JDBC statement的\_\_\_executeUpdate方法。最终页面被重定向，Command参数的值被重置为EDIT。



1. 重定向之后的请求

如图 2.3，在doPost方法中将创建一个**TableSQLData对象**以保存条件，调用TableData对象的selectEdit方法（static方法）获取数据，然后将**使用xmlEngie获取视图模板**，**向模板中注入参数和数据**，之后打印模板。



## xmlEngine

xmlEngine是Openbravo体系结构中比较有特点的一个部分，他**没有使用通常的JSP页面作为表现层（可能是由于历史原因，现在看来这种方式可能值得商榷），而是自己开发了一套基于xml的模板的引擎以渲染表现层**，可以**根据写有sql的xml文件读取数据，并将数据嵌入xml形式的HTML模板中**。

# ****编译系统结构****

## ****概述****

得益于Openbravo 较为完善的编译系统，他的一个主要的优点就是其开发速度快，能后迅速的开发出一个应用程序。

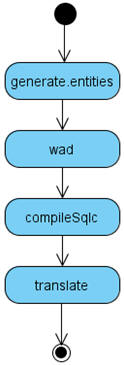
主要开发过程如下：

1. **在数据库中生成表**
2. **在Openbravo中定义表和字段**
3. **在Openbravo中定义windows，tab**
4. **编译window**

这样一个具有前端显示和后台业务数据结构的程序就生成了。

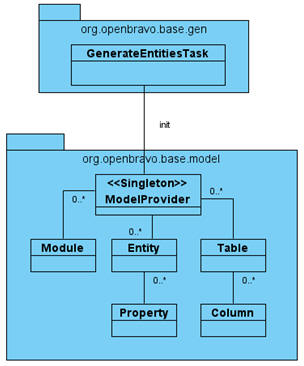
## ****编译过程****

编译任务由Ant任务**compile.development**完成。主要有以下几个子任务构成。（参见/src/build.xml）



### **generate.entities任务**

generate.entities的任务主要是**检查更新的数据字典**，在**ModelProvider中实例化org.openbravo.base.model包中的类型**，以提供给系统的其他部分使用。



**任务中的类的主要关系**

### **wad任务**

Wad任务的主要工作使用org.openbravo.wad包中的Wad类来完成任务。**Wad根据传入的参数，读取数据库信息生成代码**，如：

1. 按钮控件配置
2. 配置web.xml

和**针对windows的每个标签页生成**：

1. 含有sql操作的xsql
2. 控制器Java源文件
3. 配置参数和数据域的xml文件

### **sqlc任务**

将含有sql操作的xml编译成Java源文件，对应业务层中的负责执行SQL的类。

### **translate任务**

根据一些信息，如程序所注册的模块，对数据进行必要的转换。完成之后所有运行时需要的文件都已经生成，重启服务器就可以看到新开发的页面了。

# ****评述****

OpenbravoERP的体系结构并不算复杂，自成一体。

软件结构表现层，控制器，和业务层较为清晰。表现层xmlEngine的设计很有独到之处。应用数据字典和编译系统的设计为程序的开发和定制提供了全面的支持，使得开发者几乎不需要手工编写代码就可以自己开发程序。

当然，运行时的三层体系结构还是略显单薄，**业务层和持久层的耦合紧密**。所以其体系结构对代码重用及模块复用的支持有限，更多的是**依靠其较为完备的应用数据字典的定义来生成使用的代码**。换句话说，主要是利用引擎生成部分相似的代码的方式来解决复用问题。

Openbravo组织意识到了这种方式的不足，也在逐渐的做出改变，在2.5版本中我们见到了一些OR-Mapping的设计，更多信息可以参考最新的产品白皮书。