J2EE 平台上的 HTML 电子表格工具的设计*)

谢正良　赵建华　李宣东　郑国梁
(南京大学计算机科学与技术系　南京 210093)

摘　要　随着 Internet 的发展,Web 应用也得到了快速发展和广泛普及。同时这也给 Web 应用开发提出了新的要求。与传统应用程序开发相比,Web 应用开发具有如下特点:开发周期短,开发成本高,实现技术复杂。为了适应 Web 应用开发的新特点,开发人员需要一种可以快速开发 Web 应用的工具。本文介绍了一种 J2EE 平台上动态 HTML 表格的设计和实现。开发者可以使用该工具定义一个和数据库相联系的动态 HTML 表格。这个工具可以根据用户的定义自动生成相应的 Servlet 代码。

关键词　J2EE,实体 Beans,Web 应用,Servlet

A Design Tool for Dynamic HTML Form on J2EE Platform

XIE Zheng-Liang　ZHAO Jian-Hua　LI Xuan-Dong　ZHENG Guo-Liang
(Department of Computer Science and Technology, Nanjing University, Nanjing 210093)

Abstract　As the Internet develops, Web applications become more and more popular. At the same time, it brings also a challenge to the Web application developers. Comparing with the development of traditional applications, the development of Web applications has the following new features: short development cycle, high development cost and complicated technique. The developers need some new tools to develop Web applications quickly. This paper introduces a tool designed for dynamic HTML forms design. The developer can use this tool to define J2EE-based dynamic HTML forms, which are widely used in Web applications. The tool can generate Servlet code for the user-defined forms automatically.

Keywords　J2EE, Entity Beans, Web application, Servlet

1　引言

Web[1]是 Internet 上的一种服务,它使用超文本技术将遍布全球的各种信息资源链接起来,以便于用户的浏览。Web 的信息资源格式是丰富多样的,包括文本、多媒体、数据库、应用程序。随着 Internet 的发展,Web 应用也得到了快速发展和广泛普及,尤其是使用浏览器作为客户端的 Brower/Server 结构的 Web 应用得到了广泛的应用。

与传统应用程序开发相比,Web 应用开发具有如下特点:开发周期短,开发成本高,实现技术复杂。为了适应 Web 应用开发的新特点,开发人员需要一种可以快速开发 Web 应用的工具。

同时我们注意到,由于表格在组织数据方面具有其独特的优势,因此在很多的 Web 应用中都是使用 HTML 表格来组织和显示各种数据信息的。可以这么说,只要是具有一定规模的 Web 应用,一般都会使用到 HTML 表格。因此,提高开发者设计表格以及相关代码的效率可以有效提高 Web 应用的开发效率。

本文将介绍一种 J2EE 平台上的 HTML 表格编辑和生成工具的设计。该工具支持 HTML 表格的编辑,并能够自动生成 J2EE 平台上的 Web 应用程序。它可以极大地缩短开发周期,降低开发成本,简化实现难度。该工具适用于中小企业快速开发 Web 应用的需要。

2　相关技术介绍

J2EE[2]是用于建立服务器方应用程序的一种系统平台。它的整体架构是一个多层结构,包括:用户层、Web 层、业务层和 EIS 层。

用户层用来与用户交互,并把来自系统的信息显示给用户。J2EE 支持不同类型的用户,包括 HTML 用户,Java Applet 或 Java 应用。

Web 层产生表示逻辑,并接受客户端的用户反馈,这些用户通常是 HTML 客户端。该层由 Jsp 或 Servlet 来实现。

业务层处理系统的业务规则,一般由 EJB(Enterprise JavaBeans)来实现。

EIS 层主要包括:后台数据库系统、遗产系统、企业资源计划系统等。

J2EE 的核心是 EJB。EJB 组件可以用于封装业务逻辑,以及帮助应用程序开发者,使他们不需要担心许多系统级问题,如事务、安全性、可扩展性、并发性、通信、资源管理、持续性、出错处理。一个 EJB 组件由一组 Java 类和 XML 配置文件组成。其中 Java 类包括:一个 Home 接口、一个 Remote 接口和一个实现接口的 Bean 类。EJB 组件主要分为三种类型:会话 Bean、实体 Bean 和消息驱动 Bean。会话 Bean 用于表示工作流、业务逻辑和应用程序状态,每次由单个的客户使用。会话 Bean 又细分为两种类型:有状态的(stateful)和无状态

*)本文的研究工作受到国家自然科学基金(批准号 60203009,60233020),江苏省自然科学基金(批准号 BK2003408)和国家 973 项目(批准号 2002CB312001)的资助。谢正良　研究生,主要研究领域为软件工程。赵建华　博士,副教授,研究领域为形式化方法,软件工程,程序设计语言。李宣东　教授,博士生导师,研究领域为形式方法,模型检验。郑国梁　教授,博士生导师,研究领域为软件工程,软件开发环境。

的(stateless)。有状态会话 Bean 可以在客户访问期间保存数据,无状态会话 Bean 不能保存这些数据。实体 Bean 可以看作是数据库的面向对象的数据表示。与数据库相似,它可以同时由多个客户访问。一个实体 Bean 可以代表一组数据,比如一个顾客、一种产品或者一个帐户。消息驱动 Bean 是异步调用的,可以通过 Java Messaging Service(Java 消息服务,JMS)提供者来接受和处理 JMS 消息。典型情况下,客户会向一个专门的 JMS Queue 或者 Destination 发出一条消息,而所有订阅此目标的消息驱动 Bean 都将接受此消息。

3 工具总体介绍

我们的工具为用户提供了一种快速开发 Web 应用的方法。它通过支持表格编辑来快速生成可定制的 Web 应用程序界面;通过支持数据查询定义、数据与表格及其单元格的对应关系,为 Web 应用程序界面中的数据获取提供了准确可靠的保证。

Web 应用程序开发者(以下简称开发者)可以使用该工具来快速生成 Web 应用程序。使用该工具一般需要如下步骤。首先使用该工具为 Web 应用程序定义界面。目前,该工具仅支持由 HTML 中表格构成的界面。然后,开发者可以定义表格中数据的来源与查询条件,以及定义表格中单元格与数据的对应关系。工具将依据这些定义,生成相应的 Web 界面代码。这些代码在运行的时候,可以按照用户的定义把数据正确地显示在相应的单元格中。最后开发者使用工具的代码自动生成功能就可以得到 J2EE 平台上的 Web 应用程序了。生成的代码中,包含了 Web 应用程序的部署文件和正确运行所需的所有信息,用户不需要添加任何代码。

使用该工具来开发 Web 应用程序,可以极大地缩短开发周期,降低开发成本,同时也降低了开发人员的技术要求。该工具特别适用于中小企业快速开发 Web 应用的需要。

该工具是我们的研究组设计的 MDA[4] 系列工具中的一个。另外两个工具是:数据实体定义工具和业务逻辑定义工具。该工具与数据实体定义工具之间的关系如图 1 所示。首先开发者通过数据实体定义工具定义 Web 应用中使用的实体,并把信息导出到文件中,并生成相应的 EJB 代码和数据库描述文件。然后本工具从文件中读入实体定义信息,进行相应的界面设计和数据与界面元素的关联,并生成处理页面的 Servlet 代码。生成的代码运行时,Servlet 需要调用 EJB 代码以动态地获得数据。本文中的工具与业务逻辑定义工具之间的关系如图 2 所示。本工具把定义好的表格信息导出到表格文件中,并生成处理页面的 Servlet。业务逻辑定义工具将从表格文件中导入表格定义信息,进行相应的业务逻辑定义,并生成业务逻辑 Servlet。生成的代码运行时,首先用户访问业务逻辑 Servlet,然后业务逻辑 Servlet 根据定义的业务规则调用相应的页面生成 Servlet,并把生成的页面返回给用户。

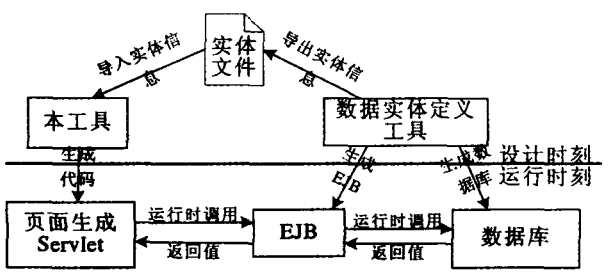

图 1

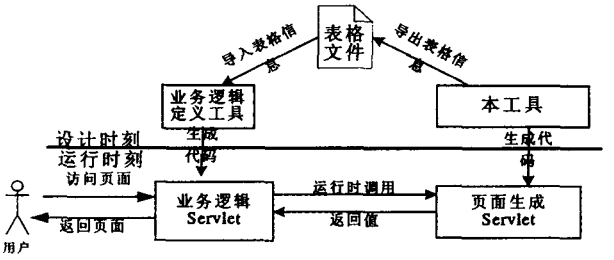

图 2

本工具的总体架构如图 3 所示。工具主要由两个部分组成:界面和程序主框架。界面部分为用户使用该工具提供了可视化的用户界面。程序主框架主要处理应用逻辑,它又细分为如下模块:数据存储模块,实体导入模块,表格导出模块,代码生成模块。

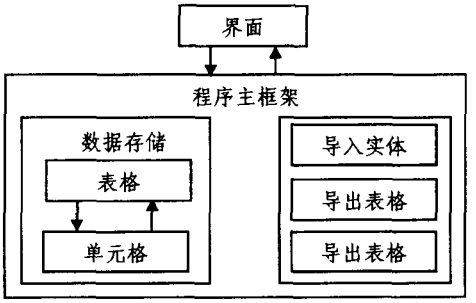

图 3

数据存储模块 它主要用来存储表格结构和表格中定义的数据信息。该模块包括表格和单元格两个类。单元格类封装了单元格的所有属性和对单元格的所有操作,是工具中最核心的类。表格类由单元格类组成,描述了单元格之间的层次关系。表格中的大部分功能都是通过调用多个单元格的相应功能来实现的。

实体导入模块 在定义表格中的数据时需要使用到实体信息。这些实体信息就是通过该模块来维护和提供的。该模块将从实体定义工具生成的实体文件中读取实体信息。

表格导出模块 该模块将把表格定义信息导出到表格文件中。业务逻辑定义工具就是从这个文件中获取表格定义信息的。

代码生成模块 该模块主要负责生成代码。生成的代码在运行时将动态地生成用户定义的界面及其界面上的数据。

4 设计和实现技术

我们将在本节中描述本工具的有关设计和实现技术,包括表格结构编辑方法,数据定义和代码生成方面的内容。

4.1 表格结构的编辑

我们的工具使用不断拆分单元格的方法来定义表格的结构。在设计表格的开始时刻,表格是一个单一的单元格。用户可以按照三种方式拆分单元格:横向拆分、纵向拆分和纵向可变拆分。通过多次拆分并设定每个单元格的大小,用户就可以定义任何形式的表格。横向拆分是在水平方向上将一个单元格拆分成为一组固定的多个单元格,单元格的数量可以由工具的使用者输入。其表格变化和数据结构的改变如图 4 所示。纵向拆分是在垂直方向上将一个单元格拆分成为一组固定的多个单元格,单元格的数量可以由工具的使用者输入。

其表格变化和数据结构的改变类似于横向拆分。纵向可变拆分是在垂直方向上将一个单元格拆分成为一组不固定的多个单元格。单元格的数量根据最终代码运行时的实际情况动态确定。比如一个单元格是纵向可变拆分的，如果在运行时查询获得的实际数据包含 n 个实体，那么该单元格就会包含 n 个子单元格。这些单元格内部的结构都是相同的。纵向可变拆分的数据结构改变如图 5 所示。我们使用树形的数据结构来记录表格的结构。在这个树型结构中，叶子节点代表了原子单元格；而非叶子节点代表了一个由其它的单元格组合而成的复合单元格。节点之间的父子关系就代表了这样的组合关系。使用这样的拆分方法的目的是方便表格中数据的定义。

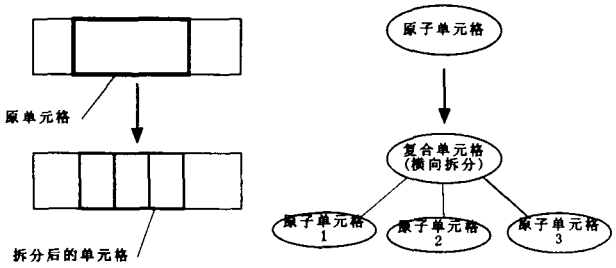

图 4

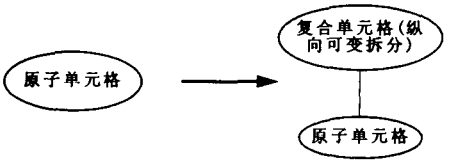

图 5

4.2 表格数据的定义

在定义了表格的基本结构之后，Web 应用开发者还需要指定表格的各个单元格中应该显示什么样的值。这样的数据定义过程被分成 3 个步骤，依次是：参数定义，实体实例定义和单元格值定义。这 3 个步骤可以交叉进行。

每个表格可以有 0 个或者多个全局参数。这些全局参数可以在定义实体实例时被引用。在 Web 应用运行的时候，全局参数的值可以通过 url 或者 session 传递给生成该页面的 Servlet。

我们的工具需要使用实体定义工具中定义的实体信息。每个实体必须至少包含一个 FindByPrimaryKey 函数，以及若干个 Find 函数。FindByPrimaryKey 函数的所有参数刚好组成实体的关键字属性。该函数的返回值是一个 EJB Remote 接口类的对象。Find 函数的参数可以不是关键字属性。此类函数的返回值是一个由 EJB Remote 接口类对象组成的集合。在本工具中定义实体实例的格式如下所示：

EntityName = (EntityType, 查找函数, 参数 1, 参数 2...)

其中 EntityName 是定义的实体实例的名字，EntityType 是实体的类型，查找函数是实体的 FindByPrimaryKey 函数或者某个 Find 函数，参数 I 对应于查找函数的第 I 个参数，参数 I 的取值可以是全局参数、常量和已定义实体实例的属性。当查找函数为 FindByPrimaryKey 的时候，我们称 EntityName 为单值实体定义。而查找函数为 Find 类型的时候，EntityName 对应于一个多值实体定义。实体实例的定义总是和某个单元格相关联。其中，单值实体可以和任意类型的单元格关联，多值实体只能和纵向可变拆分单元格关联，而且与纵向可变拆分单元格关联的多值实体数最多为一个。在 Web 应用运行的时候，纵向可变拆分的子单元格的个数就等于该多值实体定义中的实体实例的个数，并且第 i 个子单元格对应于这些实体实例中的第 i 个。

开发者需要定义表格的每个单元格的值。单元格值分为两种类型：常量和实体实例的属性。常量定义比较简单，开发者直接输入字符常量就可以了。在 Web 应用运行时该单元格的值就是这个字符常量。当单元格值是一个实体实例的属性时，它的定义格式为：

EntityName. Attribute

其中，EntityName 是与该单元格，或者该单元格的祖先关联的实体实例，而 Attribute 是 EntityName 的属性。如果 EntityName 是单值实体，在 Web 应用运行时刻，该单元格的值就是 EntityName 对应的实体实例的 Attribute 属性的值。如果 EntityName 是多值实体，那么它必然和一个被纵向可变拆分的单元格相关联。而在 Web 应用运行时刻，每个实体实例都对应于一个子单元格。在第 i 个子单元格(或者它的后代单元格)中引用 EntityName. Attribute 就是第 i 个实体的 Attirbute 属性的值。

4.3 代码生成

在用户设计完表格的结构并定义好单元格的值之后，工具的代码生成模块就可以为用户生成基于 J2EE 架构的 Web 应用源代码。最终代码的一般结构如图 6 所示。其中，公用代码部分是本工具提供的代码库。它完成一些底层的处理表格结构和定义信息的数据维护功能。这些代码将被工具生成的代码调用。工具生成的代码包括：页面生成 Servlet、表格结构和定义信息、数据查询管理器。页面生成 Servlet 主要负责与客户端的交互。每个表格对应一个页面生成 Servlet。表格结构和定义信息存放了用户所设计的界面信息。在代码运行时刻，公用代码会读取这些信息，并依据这些信息去查询数据。数据查询管理器主要负责对 EJB 的管理以及通过 EJB 查询数据。在代码运行时刻，这些代码的执行流程如下：1、首先客户端访问页面生成 Servlet，并提供相应的参数值；2、页面生成 Servlet 调用公用代码；3、公用代码读取表格结构和定义信息，并生成表格的内部数据表示。在此过程中，公用代码可以调用数据查询管理器相应的方法获得数据库中的数据；4、公用代码把表格的内部数据按照表格结构转换成 HTML 代码，并把 HTML 代码返回给页面生成 Servlet；5、页面生成 Servlet 对 HTML 代码进行适当的包装，然后返回给客户端。

在生成代码时，有关表格结构和数据定义信息被保存在工具生成的“表格结构和定义信息”中。在运行时刻，公用代码将按照表格的定义信息去从数据库中查询数据。这些查询得到的数据会被保存在一个内部结构中，并被用来生成表格页面。为此，我们在生成的源代码中定义了单元格类 class cell 来存放查询得到的数据。每个 cell 的实例对应于表格的一个单元格。一个表格所有的 cell 实例被组织成为和表格结构对应的树型结构。而单元格之间的分割包含关系被表示为 cell 实例之间的父子节点关系。本工具生成源代码所采用的基本方法是：为表格定义中的每一个单元格生成一个对应函数。该函数在 Web 应用运行时将创建一个 cell 的实例，并且按照定义计算这个单元格属性值。该函数的返回值就是它创建的单元格实例。一个复合单元格所对应的函数将调用对应

于它的每个子单元格的函数来生成它的子单元格对应的 cell 实例。这些 cell 实例将被加入到复合单元格对应的 cell 实例的孩子节点中。这样在 Web 应用运行时刻，公用代码只要调用表格的根单元格所对应的函数就可以获得表格的定义信息。比如有三个单元格 Cell1，Cell2 和 Cell3，其中 Cell1 是 Cell2 和 Cell3 的父亲节点（即，Cell1 被分割为 Cell2 和 Cell3）。那么在生成源代码时就会生成三个函数：Func_Cell1，Func_Cell2，Func_Cell3。每个函数都会创建一个类 cell 的对象，并且根据对应单元格的属性值对已经创建的单元格对象属性进行赋值。同时，Func_Cell1 函数将调用 Func_Cell2 和 Func_Cell3，并把它们的返回值加入到自己创建的单元格对象的孩子节点中。它们之间的关系如图 7 所示。

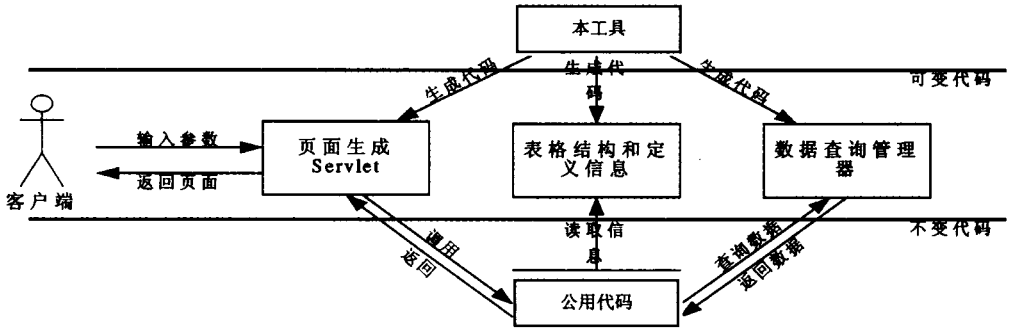

图 6

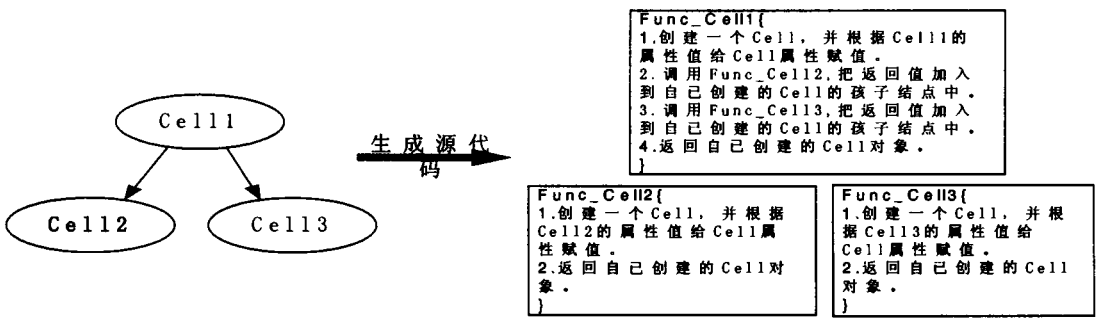

图 7

生成的代码运行时，需要通过访问 EJB 来完成数据操作。EJB 是由实体定义工具自动生成的，每个实体对应着一个 EJB 实体 Bean。如果一个表格使用到了多个不同类型的实体，那么生成的代码运行时就必须访问多个不同的 EJB 实体 Bean。由于公共代码部分适用于所有的表格定义，因此在公共代码中不可能出现具体的 EJB 的名字。也就是说，在查询数据的时候公用代码不能直接访问这些 Beans 的 Find 函数或者 FindByPrimaryKey 函数。为了解决这个问题，我们让工具为每个表格生成一个数据查询管理器。在管理器中枚举了所有的实体类型以及这种实体类型在每个查询函数下访问 EJB 时的不同情况。当公共代码根据实体定义去获取实体实例的时候，只要向数据查询管理器提供定义中说明的实体类型和相应的查找函数的名字，以及相关参数，查询管理器就会根据这些信息完成对相关 EJB 的函数调用，并返回相应的值。

总结 本文介绍了一种 J2EE 平台上的 HTML 表格编辑和生成工具的设计。该工具支持 HTML 表格的编辑和表格中数据的定义，并能够自动生成 J2EE 平台上的 Web 应用程序。它可以极大地缩短开发周期，降低开发成本，简化实现难度。该工具特别适用于中小企业快速开发 Web 应用的需要。

现在市场上有一些基于 J2EE 平台的 Web 报表制作系统。使用这类系统的一般步骤包括：首先用户定义报表的结构形式，然后定义报表中相应的单元格应该显示什么数据，最后就可以自动生成所定义的 Web 报表了。和我们的工具相比，大部分这样的系统定义的表格结构是有限制的，它只能定义出某些特定结构的表格。在这些系统中，通常一个表格只能被定义成若干行和若干列。而我们的工具在定义表格结构时是没有任何限制的，可以定义出任意结构的表格。

另一方面，我们的工作是作为我们的研究小组研究的 MDA 系列工具中的一个有机组成部分而存在的。和 MDA 中的实体定义工具以及业务逻辑定义工具结合，我们的工具可以为用户提供更加强大的功能。

参 考 文 献

1 马洪兵，张秋玲编著. HTML 语言与 Web 站点开发技术. 北京：清华大学出版社，1998

2 马树奇(译). J2EE 编程指南(1.3 版). 北京：电子工业出版社，2002

3 UML Profile for Enterprise Distributed Object Computing Specification. Available at: http://www.omg.org.

4 MDA Guide. Available at: http://www.omg.org/docs/omg/03-06-01.pdf