

面向数据绑定的表格建模研究与实现^{*})

许 晖 徐 鹏

(清华大学计算机科学与技术系 北京 100084)

摘 要 表格作为结构化数据的一种最常用的表现形式,已被各个行业广泛应用在信息管理系统中。表格的自动生成和处理技术也成为应用信息系统设计和实现中的关键技术。随着 Internet 和 Web 技术的发展,Web 与传统的数据库系统、文档管理系统结合在一起构成了巨大的、异构的、分布式的数据仓库。海量信息和异构信息的处理对表格的数据建模和处理技术提出了新的要求。针对现有技术在表格建模方面存在的模型结构简单,支持的表格类型有限和缺乏对集成异构数据源的支持等缺陷,本文在从设计和实现方面分析现有各种信息系统对表格处理技术的需求基础上,提出了一种面向数据绑定的新型表格建模方法以及相关技术,从而满足数据层和显示层在数据源类型、数据集成和组织方式等方面的多种需求。

关键词 表格,数据模型,数据绑定,XML,XPath,Java

Research and Implementation of Table Modeling Oriented to Data Binding

XU Hui XU Peng

(Department of Computer Science and Technology, Tsinghua University, Beijing 100084)

Abstract As a common used format of data representation for structured data, table has been widely applied in various information systems. The automated creation and processing technologies of table have become the key technique for most of the information systems. Nowadays, Web, together with the traditional databases and document managing systems, makes up of an enormous, heterogeneous and distributed data warehouse, coming with which the aim of processing the massive and heterogeneous information has tabled new proposals to the modeling and processing of table. However, the existing techniques on modeling of table have been less focused on various types of table and the integration of heterogeneous data source, and the structure of the current models seems a little simple. Aiming to these scarcities, this article analyses the requirements of table processing techniques in the design and implementation of various information system and proposes the table modeling method oriented to data binding, which can satisfy the requirements of the data level and the view level on data type, data integration and data organization.

Keywords Table, Data model, Data binding, XML, Java

1 引言

表格具备格式清晰、内容层次分明等显著特点,是结构化数据在图形化用户界面中进行组织、表示和操作的常见形式,被广泛使用于各行业各领域的信息系统中。在企业级文档管理系统所管理的各种商务报告和报表中,表格的使用频率非常高,且是提供信息重要性最高的资源类型。由于各个应用信息系统对所操作的内容的结构定义和表格布局方式有着不同的要求,因此通用的表格自动生成和处理技术就成为信息系统设计和实现中的一项关键技术。

在当今的信息社会中,Web 与传统的数据库系统、文档管理系统结合在一起构成了巨大的、异构的、分布式的数据仓库,信息系统需要处理异构的海量信息,但是普通的表格模型在结构上显得过于简单,而且支持的表格类型有限,又缺乏对集成异构数据源的支持,已不能满足实际处理的需要。针对实际信息处理的应用中对表格的数据建模和处理技术提出的新的要求,本文提出了针对信息自动处理目标的面向数据绑定的表格模型。该表格模型从对结构化数据的特性和数据在表格中的布局习惯出发,将表格构件的样式定义与相关内容

定义紧密结合,通过严格的形式化定义,形成了一种支持各种类型数据源绑定关系定义需求的表格建模过程描述机制,并在此基础上提出了相关的表格自动生成和处理技术。这种模型的结构描述机制结构清晰,既突出了表格内容层次分明的特点,应用范围广泛,又具备实际应用价值。

本文的后续篇幅将从三个部分对研究工作进行介绍:第 2 部分结合现有信息系统的设计,分析表格建模的需求,包括表格的应用需求和现有表格模型的分析;第 3 部分介绍面向数据绑定的表格模型的形式化定义以及实现;最后将总结本文的研究工作和未来研究方向。

2 表格建模的需求分析

2.1 表格的数据分类

从数据角度来看,表格是数据的集合体,其中的同构数据在布局中一般集中在一个矩形区域中,而异构数据的分布则无规律可寻,而表格的数据都来自特定的数据源,因此数据源的特性对表格的生成和处理有着重要影响。根据数据源结构的不同,可以将其分为两种类型:

1. 记录型数据集 来自结构化数据源,比如 XML 文档、

^{*})国际合作项目。许 晖 硕士研究生。徐 鹏 博士,助理研究员。

关系数据库,属于同构数据。同一记录型数据集中的数据项具有同构的特性。记录型数据集的数据模型形式化描述如下:

定义 1 记录型数据集的数据模式 SRD (Schema of Record Data) 是一个三元组: $SRD = (R, T, D)$, 其中 R 代表记录型数据集的根; T 代表数据集中数据项的字段特征集, $T = (type_1, type_2, \dots, type_n)$, n 是特征集的基数; D 代表记录型数据集, D 是一个有限集合, $D = \{D_i | 1 \leq i \leq m\}$, m 是记录型数据集的基数, $m = |D|$, $D_i = ((type_1, data_{i,1}), (type_2, data_{i,2}), \dots, (type_n, data_{i,n}))$ 。

记录型表格数据来源于记录型数据集,一般的应用中都采用一对一的绑定,即一个表格数据单元绑定一个记录型数据集的数据元素。

2. 非记录型数据集 由非记录型数据组成。相对于记录型数据而言,非记录型数据的数据来源比较零散,很少直接来自结构化数据源,不具备良好的结构性特征,属于异构数据。

2.2 表格的应用类型

表格在布局和数据结构定义方面有很大的灵活性。由于实际应用中的大部分表格都是建立在记录型数据集的基础上的,基于非记录型数据集的表格的内容在排列上无规律,所以本文不进行讨论。为了表达表格的结构,本文首先引入伪表格^[8]的概念。伪表格由 Seung-Jin Lim 和 Yin-Kai Ng 在 1999 年提出,其定义如下:

定义 2 一个伪表格 $T = (C, C', A)$, 其中 C 代表表格的标题, C' 为列标题构成的序列 $C_1, C_2, \dots, C_n$, A 为表格单元格数据的集合, $A = \{(a_{1,1}, \dots, a_{1,n}), (a_{2,1}, \dots, a_{2,n}), \dots, (a_{m,1}, \dots, a_{m,n})\}$ 。伪表格 T 是一个二维表格, 其中每个列标题 C_i ($1 \leq i \leq n$) 或单元格数据 $a_{i,j}$ ($1 \leq i \leq n, 1 \leq j \leq m$) 可以为空。

在不同的应用中,表格有着不同的布局和数据组织方式。下面将从信息组织方式和标题布局方式两个方面对现有的表格类型进行分类。

1. 面向信息组织方式的分类方法

(1) 行优先的表格。行优先的表格以行为单位组织数据,一般每一列的数据具有相同的数据结构。这种类型的表格可以按照如下的形式化描述映射成伪表格:

定义 3 一个行优先的表格 $T = \{(h_{1,1}, \dots, h_{1,n}), \dots, (h_{k,1}, \dots, h_{k,n}), (d_{k+1,1}, \dots, d_{k+1,n}), \dots, (d_{k+m,1}, \dots, d_{k+m,n})\}$, 其中每个 $h_{i,j}$ ($1 \leq i \leq k, 1 \leq j \leq n$) 代表表格中属于标题性质的单元格的数据内容, 每个 $d_{i,j}$ ($k+1 \leq i \leq k+m, 1 \leq j \leq n$) 代表数据区中单元格的数据内容。

(2) 列优先的表格。列优先的表格以列为单位组织数据,一般每一行的数据具有相同的数据结构。从空间上看,这种类型表格的标题区和后续数据区构成的矩阵可以通过矩阵变换操作并重组后转换为行优先的表格。列优先的表格可以按照如下的形式化描述映射成伪表格:

定义 4 一个列优先的表格 $T = \{(h_{1,1}, \dots, h_{1,k}, d_{1,k+1}, \dots, d_{1,k+m}), \dots, (h_{n,1}, \dots, h_{n,k}, d_{n,k+1}, \dots, d_{n,k+m})\}$, 其中每个 $h_{i,j}$ ($1 \leq i \leq n, 1 \leq j \leq k$) 代表表格中属于标题性质的单元格的数据内容, 每个 $d_{i,j}$ ($1 \leq i \leq n, k+1 \leq j \leq k+m$) 代表数据区中单元格的数据内容。

在实际应用中还经常用到一种表格,其顶部具有至少一个标题行,最左部具有至少一个标题列。从数据组织结构来看,这种表格的每一行和每一列的数据之间都具有相同的结

构,所以这种表格既可以看成是行优先的表格,也可以看成是列优先的表格。

2. 面向标题的分类 标题是表格的重要组成部分,一般的表格都可以分为标题和数据两部分。按照标题和数据的组织结构的不同,可以将表格分成如下两种类型:

(1) 标题与数据分离的表格。从几何角度分析,这种类型表格的标题位于独立的区域而与数据区分离,以行标题或列标题的方式和数据区一起构成表格。

对于独立的标题区,很多实际应用常采用分层分组的嵌套结构,通过将标题区分层、子标题分组的方式表示某一数据类型的细分结构。图 1 是上市公司年报中采用分层分组结构定义表格标题区的实例。

项目	股票投资收益		债券投资收益	其他投资收益		合计
	成本法	权益法		成本法	权益法	
短期投资						
长期投资						
合 计						

图 1 分层分组的标题

行优先的分层分组标题表格可以通过如下的形式化描述映射成伪表格:

定义 5 一个行优先的标题区为分层分组结构的表格

$$T = \{(h_{1,1}, \dots, h_{1,n_1}), (h_{2,1}, \dots, h_{2,n_2}), \dots, (h_{k,1}, \dots, h_{k,n_k}), (d_{k+1,1}, \dots, d_{k+1,n_k}), \dots, (d_{k+m,1}, \dots, d_{k+m,n_k})\}$$

$1 \leq n_1 < n_2 < \dots < n_k$, 其中每个 $h_{i,j}$ ($1 \leq i \leq k, 1 \leq j \leq n_i$) 代表表格中属于标题性质的单元格的数据内容, 每个 $d_{i,j}$ ($k+1 \leq i \leq k+m, 1 \leq j \leq n_k$) 代表数据区中单元格的数据内容。

列优先的分层分组标题表格可以通过对定义 5 的单元格矩阵作矩阵操作映射为伪表格。

(2) 标题和数据相间的表格。这种类型的表格的特点是标题和数据间隔出现,表格的行或列由一个或多个多元组(标题,数据 1,数据 2, ..., 数据 k)构成。图 2,3 显示了两种该类型表格的组织方式。

标题 1	标题 2
标题 3	标题 4

图 2 标题区和数据区间隔的表格——标题行方式

标题 1		标题 3	
标题 2		标题 4	

图 3 标题区和数据区间隔的表格——标题列方式

以行优先的这类表格为例,其可以通过如下的形式化描述映射成伪表格:

定义 6 一个行优先的标题与数据相间的表格, $T = \{(h_{1,1}, d_{1,1,1}, \dots, d_{1,1,k_1,1}, \dots, h_{1,n_1}, d_{1,n_1,1}, \dots, d_{1,n_1,k_1,n_1}), \dots, (h_{m,1}, d_{m,1,1}, \dots, d_{m,1,k_m,1}, \dots, h_{m,n_m}, d_{m,n_m,1}, \dots, d_{m,n_m,k_m,n_m})\}$, 其中每个 $h_{i,j}$ ($1 \leq i \leq m, 1 \leq j \leq \max\{n_1, \dots, n_m\}$), 代表表格中属于标题性质的单元格的数据内容, 每个 $d_{i,j,l}$ ($1 \leq i \leq m, 1 \leq j \leq n_i, 1 \leq l \leq k_{i,j}$) 代表数据区中单元格的数据内容。

列优先的这种类型的表格可以通过对定义 6 的单元格矩

阵作矩阵操作映射成为表格。

2.3 现有表格模型的分析

为了设计表格构件处理表格数据,需要对表格进行建模,绝大多数软件开发工具包都设计了相应的表格模型,并提供了表格构件以供开发人员使用。MVC (Model-View-Controller) 设计模式是现有表格构件经常采用的结构。MVC 结构将一个交互式应用分为三个层次:M 层包含核心功能和数据,V 层用于实现信息的显示功能,C 层负责控制和处理用户输入。MVC 的优点在于能够将数据与显示和处理相分离,独

立出系统的数据模型。

Java 平台开发工具包 (Java Develop Kit, JDK) 的 GUI (Graphical User Interface) 开发包 Swing 提供了表格构件 JTable。在结构上 JTable 采用了 MVC 模式,将表格的数据模型独立出来,在表格模型的设计中采用了以列为单位的模型设计,由列模型管理整个表格模型,在标题的处理上,将标题区和数据区分开处理。图 4 是 JTable 构件获取数据的序列图。

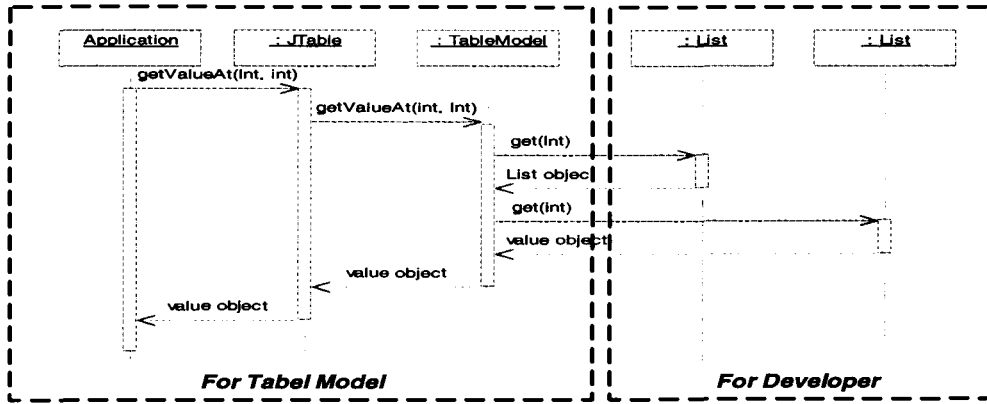


图 4 JTable 构件获取数据的序列图

从序列图可以看出,JTable 构件是通过二维数组的方式直接组织数据的,在与数据源集成方面,大部分数据组织和维护的工作还需要开发者来完成。

SWT(Standard Widget Toolkit) 是集成本机窗口系统的构件集和图形库,提供了独立于操作系统的 API,是目前比较

流行的 Java 图形界面开发工具包。SWT 中也提供了表格构件,但没有采用 MVC 设计模式,数据模型和显示层是耦合的,其表格模型也是采用了以列为单位的模型设计。图 5 是 SWT 表格构件获取数据的序列图。

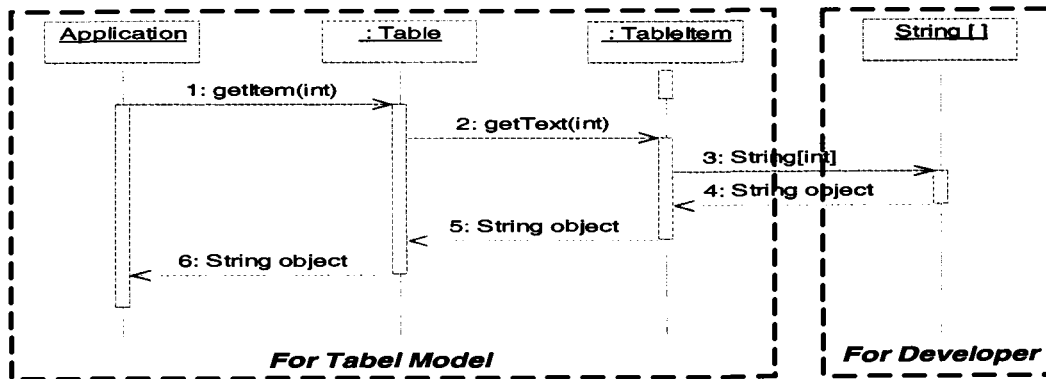


图 5 SWT 表格构件获取数据的序列图

从序列图可以看出,SWT 的表格构件也是直接通过简单数组方式组织数据的,而且只支持 String 类型的数据,需要开发者实现对数据源的集成和对复杂数据类型的支持。

Swing 和 SWT 所提供的表格构件的模型可以统一地用如下的形式化描述:

定义 7 表格数据模型 $TM = (TC, TD)$, 其中 TC 是一个 N 元组,表示表格的标题区, $TC = (C_1, C_2, \dots, C_n)$, n 是标题数; TD 是 $m \times n$ 的矩阵,表示表格的数据区, m 是表格数据模型中记录的条数。

从上述分析可以看出,虽然 Swing 和 SWT 的表格模型都为开发者提供了一套组织表格数据的接口,但是对于日益广泛和灵活的表格应用需求而言,模型的结构显得过于简单,

只能描述单一结构的表格类型,在很多方面不能满足实际需求,需要开发者扩展,从而增加了开发者的负担。下面列出了这两种模型存在的主要的不足之处:

(1)这两种模型都采用由列模型管理整个模型的结构,在默认情况下以行为单位显示记录,每一列对应同一类型的数据,没有在模型层提供支持列优先类型表格的接口,需要开发者实现对列优先表格的支持;

(2)这两种模型都采用了标题区和数据区分离的模型,没有在模型层提供支持标题和数据相间的表格的接口,需要开发人员对模型进行扩展;

(3)这两种模型都直接以数组的形式组织数据,在实际应用中需要开发者对数据源进行转换。数据源的结构信息没有

充分利用,需要开发者处理记录型数据集和非记录型数据集;

(4)这两种模型都没有提供支持数据绑定的接口,虽然可以由开发者对模型进行扩展,但增加了开发者的负担;

(5)这两种模型默认的标题区都以单行出现,没有提供支持分层分组的标题的接口,需要开发者扩展表格模型实现。

3 面向数据绑定的表格模型

针对现有表格模型的缺陷,本文从数据源类型、数据集成和内容组织方式三个方面,研究了表格建模和描述机制,提出了面向数据绑定的表格模型,通过 Xpath^[2] 定义数据源和表格的绑定,并给出了模型的形式化定义。和现有表格模型相比,本文提出的表格模型功能更完备,可以大幅度减少开发人员的工作量。

在数据源类型方面,本文提出的模型引入了记录型数据集和非记录型数据集的定义,支持和记录型数据集的绑定,从而能够方便开发者实现表格的自动生成。

在数据集成方面,本文提出了面向数据绑定的建模方式。数据源的数据模型对于信息系统而言十分重要,不同的应用一般会采用数据模型截然不同的数据源,例如关系数据库、XML 文档、对象树结构等等。表格的数据一般都来自数据源。通过数据绑定定义,可以实现表格模型和各种异构数据源的集成。本文在表格模型定义中规定采用统一、简单的数据访问接口——XPath^[2] 定义数据绑定。所有可以直接或间接通过 Xpath 访问的数据模型,例如 XForms^[3] 规范中的数据模型,IBM 和 BEA 公司联合提出的服务数据模型 (Service Data Object, SDO)^[4],都可以采用 XPath 作为访问接口与本文的表格模型集成。

在内容组织方式方面,本文提出的表格模型支持多种不同类型的表格组织方式,包括上文分析的行优先的表格、列优先的表格、标题与数据分离的表格、标题与数据相间的表格以及具有分层分组标题的表格。

3.1 表格模型的形式化定义

从总体上看,表格根据数据类型可以分为三种类型:记录型表格、非记录型表格和混合型表格。三种表格类型的不同之处在于数据定义的不同。下面的形式化定义完整地描述了本文提出的面向数据绑定的表格模型。

定义 8 已知 n 元组 $C = (c_1, c_2, \dots, c_n)$, m 元组 C'_m 满足 $C'_m = ((c'_1, (c_1, \dots, c_{i_1})), (c'_2, (c_{i_1+1}, \dots, c_{i_1+i_2})), \dots, (c'_{m-1}, (c_{\sum_{k=1}^{m-1} i_k+1}, \dots, c_n)))$, 其中 $i_1 + i_2 + \dots + i_m = n, m = 1, m < n$, 即 $i_1, i_2, \dots, i_m$ 是 n 的一个 m 整数划分。 D 为 $\{C\} \rightarrow \{C'_m \mid m = 1, m < n\}$ 上的一个映射,即 $D(C) = C'_m$ 。 $\forall C'$, 若 C' 满足 $D(C) = C'$, 则称 C' 是 C 的一个划分。

定义 9 已知字段数为 n 的记录型表格 $T, T = (TC, C, CG, RDS, DR)$, 其中 TC 表示表格的标题; n 元组 C 表示字段标题元组, $C = (c_1, c_2, \dots, c_n)$; CG 表示字段标题分层分组区,

$CG = \begin{bmatrix} C_1 \\ \vdots \\ C_{m_c} \end{bmatrix}$, 其中 m_c 为字段标题分层分组区的层数, C_{m_c} 是

C 的一个划分, C_{i-1} 是 C_i 的一个划分, $1 < i \leq m_c$; RDS 表示和记录型数据集绑定的表格数据集的集合, $RDS = \{(root_i, binding_i, start_i, offset_i) \mid i = 1, 2, \dots, m_r\}$, 其中 m_r 为 RDS 中表格数据集的个数, $root_i$ 为第 i 个表格数据集的数据绑定的根节点,用 Xpath 表示, $binding_i$ 为第 i 个表格数据集中记录

的数据绑定定义, $binding_i = (binding_{i,1}, binding_{i,2}, \dots, bind-$

$ing_{i,n})$ 或 $\begin{bmatrix} (c_{1,1}, c_{1,2}, \dots, c_{1,n}) \\ (binding_{1,1}, binding_{1,2}, \dots, binding_{1,n}) \\ \vdots \\ (c_{k_i,1}, c_{k_i,2}, \dots, c_{k_i,n}) \\ (binding_{k_i,1}, binding_{k_i,2}, \dots, binding_{k_i,n}) \end{bmatrix}$, 分别表

示单条记录在表格中以单行(列)和多行(列)的组织方式, $start_i$ 为第 i 个记录集的数据绑定的起始节点, $offset_i$ 为起始节点和终止节点之间的偏移量; DR 表示表格的行(列)优先属性, $DR \in \{row, column\}$, $DR = row$ 表示 T 为行优先表格, $DR = column$ 表示 T 为列优先表格。

定义 10 已知字段数为 n 的非记录型表格 $T, T = (TC, C, CG, NRDS, DR)$, TC, C, CG 和 DR 的定义与定义 2 相同; $NRDS$ 表示非记录型数据集, $NRDS = \{(data_{i,1}, data_{i,2}, \dots, data_{i,n}) \mid i = 1, 2, \dots, m_w\}$, 其中 m_w 表示 $NRDS$ 中非记录型数据项的数量, $data_{i,k}$ 表示 $NRDS$ 中第 i 条非记录型数据项第 k 个字段的数值, $data_{i,k}$ 可以为静态数据或者 Xpath 格式的数据绑定定义。

定义 11 已知字段数为 n 的混合型表格 $T, T = (TC, C, CG, RDS, NRDS, DR)$, $TC, C, CG, RDS, NRDS$ 和 DR 的定义与定义 9 和定义 10 中相同。

在本文提出的模型包括了第 2 部分所分析的各种表格类型。定义 9、10、11 中通过 DR 定义了表格的行(列)优先类型,从而实现了行(列)优先类型表格的支持。在模型中,所有的表格都由标题区和数据区两部分组成,其中标题区可能为空。定义 8 给出了划分概念的定义,通过划分给出了标题的分层和分组的形式化定义,从而在模型中支持这种类型的标题的定义。定义 9 中的 RDS 定义分别定义了标题和数据分离或相间的表格类型。对于第 2 部分所分析的表格类型,定义 9、10、11 和第 2 部分所给出的形式化定义是等价的。

定义 9 给出了记录型表格的形式化定义。记录型表格的数据区由结构化的记录型数据构成。本文在定义中引入了数据绑定的概念,以 Xpath 作为数据绑定的方式,通过对记录型数据集的根节点和子节点的绑定定义,完成整个数据集和表格模型的绑定,从而完成表格模型和异构数据源的集成。记录型数据在表格中有两种组织方式,单记录单行(列)和单记录多行(列)。定义 9 中在记录的数据绑定定义中给出了这两种组织方式的定义。

定义 10 给出了非记录型表格的形式化定义。非记录型数据是松散的,没有结构化特征,所以在定义中采用了松散的定义,区别于定义 9 中表格模型和数据模型之间在结构上的紧密耦合。同时,单个非记录型数据既支持静态数据定义,也支持通过 Xpath 的动态数据绑定定义,从而能够增加数据定义的灵活性,支持异构数据源。

混合型表格是记录型表格和非记录型表格的结合。定义 11 给出了这种表格的形式化定义。和上述两种类型表格的区别在于,混合型表格的数据区同时包含记录型数据和非记录型数据。本文提出的模型将记录型和非记录型数据分开,分别进行处理。

3.2 表格模型的实现

对于同一模型的实现方式可能有多种。本文针对所提出的面向数据绑定的表格模型给出了一种实现,图 6 是实现方

(下转第 62 页)

