基于数据源向图的数据库设计中数据关系的表示工具

陈冰川[1] 陈蔼祥[1] 吴向军[2] 李 磊[3]
(广东财经大学数学与统计学院 广州 510320)[1] (中山大学软件学院 广州 510275)[2]
(中山大学软件研究所 广州 510275)[3]

摘 要 数据库设计是信息系统需求分析到系统实现中的一个关键环节。传统的数据库设计方法需根据需求分析阶段的结果,依赖人的构造性思维抽象出对象和关系,由于对所需数据结构和关系的描述能力不足,缺少对数据项之间关系的描述,导致数据关系不明确,数据库设计结果容易出现偏差。在新的数据库设计工具——数据源向图的帮助下,对象、关系以及各数据项之间的关系能被直观、简洁、准确地表示,同时其也能极大地消除各种关系不明确而带来的二义性问题,可有效提高信息系统实现的效率。

关键词 数据库设计,数据源向图,有向图
中图法分类号 TP311.13 **文献标识码** A

Representation Tool of Data Relations in Database Design Based on Data Source-target Digraph

CHEN Bing-chuan[1] CHEN Ai-xiang[1] WU Xiang-jun[2] LI Lei[3]
(School of Mathematics and Statics, Guangdong University of Finance and Economics, Guangzhou 510320, China)[1]
(School of Software, Sun Yat-sen University, Guangzhou 510275, China)[2]
(Software Research Institute, SunYat-sen University, Guangzhou 510275, China)[3]

Abstract Database design is a key step between requirement analysis and system implementation in information system engineering. According to the result of requirement analysis, traditional approaches of database design utilize human constructive thinking to abstract objects and their relationships. Since traditional approaches have some defects in describing data structures and relationships, especially for relationships between data items, the results of these approaches may deviate from the real situation. In this paper, we presented a new database design tool—data flowing direction graph (DFDG) to represent objects, relationships and relationships between data items. Our cases show that results of DFDG enjoy more simple, clear, correct and unambitious than those of former methods and can improve the correctness and reliability of information system while shortening implementing time needed.

Keywords Database design, Data source-target digraph, Directed graph

1 引言

数据库设计是软件开发周期中承前启后的重要一环,其重要作用是毋庸置疑的,因此这方面的工作很多。

本质上,数据库设计就是要根据需求分析阶段的结果,为现实世界的数据及关系与计算机内编码之间建立简洁而准确的形式映射,方便后续的编码实现工作。

标准的数据库设计主要分为需求分析、逻辑设计和物理设计 3 个阶段[1]。当前普遍的数据库设计方法都是跳跃式的,凭直觉构建[2]。数据库设计中最重要的方法就是 ER 模型[3],它首先根据需求分析的结果抽象出对象和对象之间的关系,然后形成 ER 图,最后根据 ER 图导出数据库的详细设计,整个过程中对象和关系的提取依赖实施人员的需要构造性思维。

根据 ER 图的构建可以知道,它明确了实体之间的对应关系,但是没有明确实体项之间的来源关系和逻辑推导关系。因为 ER 图中连线两端的数字表示实体之间的连接关系和实体与关系之间的对应关系,所以这种表达方式无法清晰地表达出项之间的逻辑关系。

事实上,无论是实体还是关系,本质上都是数据的集合。本文提出的数据源向图的方法,是在需求获取与分析中使用数据源向图[4]的基础上发展而成的,目的就是要以图形的方式来说明数据项之间的来源和逻辑关系。本文工作主要集中在需求分析和逻辑设计两个阶段。

具体地,针对 ER 图存在的问题,本文试图通过数据源向图的建立达到如下两个目的,使软件开发过程中数据库的设计过程更加平滑和易于理解。

1)反映出各数据项之间的逻辑关系。

2)概要设计能够平滑地从需求过渡得到设计,减少构造性思维。

本文的主要贡献包括:将数据源向图进行了形式化表示,结合在需求分析中的应用为后续自动地由需求到数据库设计

本文受国家自然科学基金国际合作与交流项目(中德合作)(61111130183),教育部重点项目(210257)资助。
陈冰川(1975—),男,博士,讲师,主要研究方向为软件工程、数据库与知识库、机器学习,E-mail:chbingch@mail2.sysu.edu.cn;陈蔼祥(1978—),男,博士,副教授,主要研究方向为智能规划、机器学习。

奠定了基础；通过数据源向图工具使得从需求到设计的过渡更加平滑，减少了跳跃性思维；使得数据库各个字段之间的关系更加清晰。

2　相关工作

数据库模型由早期层次模型(Hierarchical Model)、网状模型(network model)发展到关系模型(Relational Model)[5-6](Codd. E. F)，为数据库实现、存储和查询带来了质的发展。但是关系模型由于是通过二维表来表现现实世界中各个对象之间的关系的，因此在表现形式上不像层次模型和网状模型能更直观地表现出数据间的关系。Chen 于 1976 年提出的 ER(实体关系)模型[3]能够通过图形的方式(ER 图)很好地体现出实体之间的关系，所以得到了广泛的应用。ER 图模型是一个伟大的结果。

目前有大量基于 ER 模型展开的工作：在 ER 模型提出之初，有很多基于 ER 模型进行扩展的[7-8]方法；随着人工智能的发展，产生了大量的自动数据库设计[9-11]方法，还有很多学者提出了便于应用的数据库设计工具[12-13]等。

ER 模型的扩展主要是弥补了 ER 模型的一些不足，增强了其描述能力，使之能更适应各种特殊问题，这些改进是对基础进行完善，还是没有解决跳跃性和数据项关系描述缺失的问题。

自动数据库设计，主要思考点也是 ER 模型的跳跃性及其难度，因此采用了机器替代人的方式，自动形成数据库设计，主要是基于学习、专家系统、语义分析等方面进行的，通过大量的实例或者人工交互得到最终的 ER 模型，从而解决了跳跃性的问题。但是在实际工程中，不是每个部门都可以给出足够多的用于学习的实例数据，因为这可能涉及到数据安全的问题；同时由于其是基于 ER 模型的，因此数据项之间的关系描述缺失的问题还是没有得到解决。

基于各种工具的数据库设计也是介于自动与手动设计之间的方法，主要是通过各种工具将手工设计过程中相对机械化的操作通过工具来完成，所以其并未改变 ER 模型设计的本质。

另外还有一些不是基于 ER 模型进行的数据库设计，如面向目标的数据库设计[1,14]，该方法将数据库设计提前到需求分析中，这就使得需求与设计更加接近了，跳跃性得到降低，但是还是无法清晰地表明数据库表项之间的关系。

本文作者早期提出在需求阶段的数据获取和分析时采用基于数据源向图的方法，使得数据获取与分析可按照一定程式化的过程进行。而数据需求分析后的工作则是数据库设计，因此进而针对当前的数据库设计无法从需求平滑过渡到设计以及不能清晰地体现数据项之间关系的问题，改进数据源向图方法，进而提出了新的数据库设计方法——基于数据源向图的数据库设计方法。

3　数据源向图

在数据库设计中，为表示数据之间的关系并消除冗余关系，本文提出了一种新的数据关系表示形式——数据源向图(Data Source-Target Digraph，DSTD)。数据源向图是一种有向图，图中结点是表中的项和说明，有向边表示结点之间的数据来源关系。通过对图中元素的处理，可消除数据库设计中的冗余关系，完成数据库的概念设计。

3.1　数据源向图的基本概念

为形式化描述数据源向图的各组成部分，下面从表的最小单位逐层给出相关概念的定义。

定义 1　项 I(Item)是一个二元组$\langle iname, f\rangle$，$iname$ 是项的名称，$f\in\{K, \text{NULL}\}$是项性质的标记，K 是关键字标识，NULL 表示无标识。

定义 2　表 T(Table)是一个二元组$\langle tname, SI\rangle$，$tname$ 是表名，$SI=\{I_1, I_2, \cdots, I_n\}$是项的集合(Set of Item)，$I_j=\langle iname_j, f_j\rangle$，$j=1,\cdots,n$。

为方便叙述，用 $iname$ 表示项$\langle iname, f\rangle$，用 $tname$ 表示表$\langle tname, SI\rangle$，用符号 $T.I_j$ 表示表 T 的第 j 个项，$GetTable(I)$表示项 I 所在的表，如：$GetTable(T.I_j)=T$。

下面用数据源向图 DSTD 来表示所有表项之间的数据来源关系。

定义 3　假设有 n 个表$\langle T_i, SI_j\rangle$，$SI_j=\{I_{j1}, I_{j2}, \cdots, I_{jm}\}$，$j=1,\cdots,n$。其数据源向图 $G_0=\langle V_0, E_0\rangle$为有向图。

(1)$V_0=SI=\bigcup\limits_{j=1,\cdots,m} SI_j$；

(2)$E_0=\{\langle v_i, v_j\rangle \mid v_i, v_j\in V_0\}$，$\langle v_i, v_j\rangle$是连接表项之间的有向边，$v_i$ 是 v_j 的源，v_j 是 v_i 的目标。

假设有 3 个表$\langle T_1, SI_1\rangle$，$\langle T_2, SI_2\rangle$和$\langle T_3, SI_3\rangle$，$SI_1=\{\langle Item_{11}, K\rangle, \cdots, \langle Item_{1s}, \text{NULL}\rangle, \cdots\}$，$SI_2=\{\langle Item_{21}, K\rangle, \cdots, \langle Item_{2t}, \text{NULL}\rangle, \cdots\}$，$SI_3=\{\langle Item_{31}, K\rangle, \cdots, \langle Item_{3w}, \text{NULL}\rangle, \cdots\}$。

在这 3 个表中，其项之间的部分关系为：$Item_{32}=Item_{11}$，$Item_{3w}=Item_{1s}+Item_{2t}$。

由定义 3 可知：

$V_0=SI_1\cup SI_2\cup SI_3$，$E=\{\langle Item_{11}, Item_{32}\rangle, \langle Item_{1s}, Item_{3w}\rangle, \langle Item_{2t}, Item_{3w}\rangle\}$。

因此，对表 T_1，T_2 和 T_3，由该部分关系所得到的数据源向图 G_0 如图 1 所示。

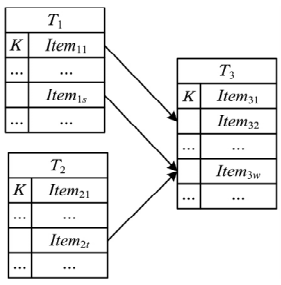


图 1　T_1，T_2 和 T_3 之间部分关系的数据源向图 G_0

由图 1 可知：项 $T_3.Item_{32}$ 的值来源于项 $T_1.Item_{11}$；项 $T_3.Item_{3w}$ 的值来源与项 $T_1.Item_{1s}$ 和 $T_2.Item_{2t}$ 有关，但它们之间的具体关系无法在图中反映。为表达表项之间的各种复杂关系，本文引入说明 D(Description)的概念。

定义 4　说明 D(Description)是一个二元组$\langle dname, dsc\rangle$，$dname$ 是说明的名称(或标记)，dsc 是说明的内容。

为方便叙述，用 $dname$ 表示说明$\langle dname, dsc\rangle$，这种简化的表达方式与前面项和表的表达方式相似。在数据源向图中引入说明后，把定义 3 修改成定义 5 的形式。

定义 5　假设有 n 个表$\langle T_i, SI_j\rangle$，$SI_j=\{I_{j1}, I_{j2}, \cdots, I_{jm}\}$，$j=1,\cdots,n$。说明集 $D=\{d_1, d_2, \cdots, d_i\}$，其数据源向图 $G=\langle V, E\rangle$为有向图，且：

(1)$V=SI\cup D$;

(2)$E=E^0\cup E^1\cup E^2$,其中:

(2.1)$E^0=\{\langle v_s,v_t\rangle\mid v_s,v_t\in SI\}$,$\forall\langle v_s,v_t\rangle\in E^0$ 表示项 v_t 的取值来源于 v_s;

(2.2)$E^1=\{\langle v_s,d_t\rangle\mid v_s\in SI,d_t\in D\}$,$\forall\langle v_s,d_t\rangle\in E_i^1$ 表示说明 d_t 中引用项 v_s 的值;

(2.3)$E^2=\{\langle d_t,v_s\rangle\mid v_s\in SI,d_t\in D\}$,$\forall\langle d_t,v_s\rangle\in E_i^2$ 表示 v_s 的值来源于说明 d_t。

对于定义 3 中的数据源向图 $G_0=\langle V_0,E_0\rangle$,由定义 5 可知:

(1)$V_0=SI\cup D$,说明集 $D=\varnothing$;

(2)$E_0=E_0^0\cup E_0^1\cup E_0^2$,其中 $E_0^0=\{\langle v_i,v_j\rangle\mid v_i,v_j\in SI\}$,$E_0^1=E_0^2=\varnothing$。

由于本文的研究以数据源向图为基础,因此基于图论中的概念给出下列定义。

定义 6 有向图 $G=\langle V,E\rangle$ 中,$\forall v_i\in V$:

(1)v_i 的出边集 $E_G^+(v_i)=\{\langle v_i,v_j\rangle\mid\langle v_i,v_j\rangle\in E\}$,其出度 $Deg^+(v_i)=|E_G^+(v_i)|$;

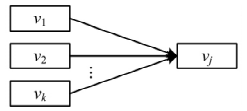
(2)v_i 的入边集 $E_G^-(v_i)=\{\langle v_j,v_i\rangle\mid\langle v_j,v_i\rangle\in E\}$,其入度 $Deg^-(v_i)=|E_G^-(v_i)|$;

(3)图 G 中的最大入度 Δ^- 表示为 $Max(\{Deg^-(v_i),v_i\in S\})$;

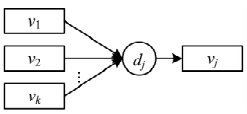
(4)v_i 的先驱集 $\Gamma_G^-(v_i)=\{v_j\mid\langle v_j,v_i\rangle\in E,v_i\neq v_j\}$。

3.2 引入说明的原理

在数据源向图 G 中,若 $v_i\in SI$,$Deg^-(v_j)>1$(如图 2(a)),则结点 v_j 值的获得与结点 $v_1,v_2,\cdots,v_k$ 的值相关。这时,引入说明〈d_j,"结点 v_j 值的获得来源于结点 $v_1,v_2,\cdots,v_k$ 的值"〉,并通过"引入说明的规则"进行操作,使之变为如图 2(b)所示的关系。



(a)$Deg^-(v_j)>1$



(b)$Deg^-(v_j)=1$

图 2 引入说明的变化示意图

引入说明的规则:在数据源向图 $G_i=\langle V_i,E_i\rangle$ 中,$V_i=SI\cup D_i$,$D_0=\varnothing$,$i\geqslant 0$,若 $v_t\in SI$,且 $Deg^-(v_t)>1$:

(1)$D_{i+1}=D_i\cup\{d_t\}$;//增加一个说明 d_t

(2)$E_{i+1}^1=E_i^1\cup\{\langle v_s,d_t\rangle\mid\langle v_s,v_t\rangle(E_{G_i}^-(v_t))\}$;//增加以 d_t 为终点、以 $E_{G_i}^-(v_t)$ 中所有始点为始点的边集

(3)$E_{i+1}^2=E_i^2\cup\{\langle d_t,v_t\rangle\}$;//添加有向边〈$d_t,v_t$〉,表示 v_t 的值是由说明 d_t 的含义得到

(4)$E_{i+1}^0=E_i^0-E_{G_i}^-(v_t)$//删除所有在 G_i 中以 v_t 为终点的边

对图 1 所示的数据源向图 G_0 利用引入说明的规则后,可得到数据源向图 G_1,如图 3 所示。

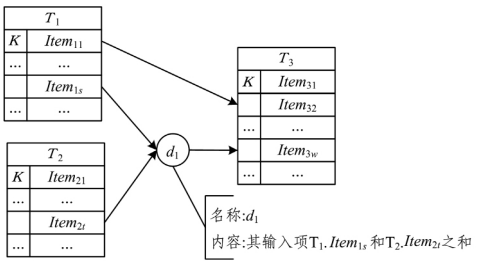


图 3 数据源向图 G_1

为方便描述,把引入说明的规则简称为引入规则。在数据源向图 $G_i=\langle V_i,E_i\rangle$ 中运用引入规则后,所得到的数据源向图称为 $G_{i+1}=\langle V_{i+1},E_{i+1}\rangle$。

假设数据源向图 $G_i=\langle V_i,E_i\rangle$,$V_i=SI\cup D_i$,$i\geqslant 0$,$v_t\in SI$,$Deg^-(v_t)=k>1$,则用 $R_{v_t}(G_i)$ 表示对结点 v_t 应用引入规则所得到的数据源向图 G_{i+1}。

性质 1 $G_{i+1}=R_{v_t}(G_i)=\langle V_{i+1},E_{i+1}\rangle$ 具有下列性质:

(1)$D_{i+1}=D_i\cup\{d_t\}$,$V_{i+1}=SI\cup D_{i+1}$;

(2)$E_{i+1}=E_i\cup\{\langle v_s,d_t\rangle\mid\langle v_s,v_t\rangle\in E_{G_i}^-(v_t)\}\cup\{\langle d_t,v_t\rangle\}-E_{G_i}^-(v_t)$;

(3)$Deg^-(v_t)=1$;

(4)$Deg^-(d_t)=k$,$Deg^+(d_t)=1$;

(5)$\forall v_s\in SI-\{v_t\}$,都有:$E_{G_{i+1}}^-(v_s)=E_{G_i}^-(v_s)$。

证明:性质(1)－性质(3)可直接由引入规则得到。

性质(4)的证明如下:

由引入规则可知:以 d_t 作为终点的边集为 $\{\langle v_s,d_t\rangle\mid\langle v_s,v_t\rangle\in E_{G_i}^-(v_t)\}$,以 d_t 作为始点的边集为 $\{\langle d_t,v_t\rangle\}$,所以,$Deg^-(d_t)=k$,$Deg^+(d_t)=1$。

性质(5)的证明如下:

假设 $v_s\in SI-\{v_t\}$,由于数据源向图 G_{i+1} 是对图 G_i 利用结点 v_t 运用引入规则所得到的,因此其边集变化为:

$$E_{i+1}=E_i\cup\{\langle v_s,d_t\rangle\mid\langle v_s,v_t\rangle\in E_{G_i}^-(v_t)\}\cup\{\langle d_t,v_t\rangle\}-E_{G_i}^-(v_t)$$

由定义 6 可知:

$$\begin{aligned}E_{G_{i+1}}^-(v_s)&=\{\langle v_j,v_s\rangle\mid\langle v_j,v_s\rangle\in E_{i+1}\}\\&=\{\langle v_j,v_s\rangle\mid\langle v_j,v_s\rangle\in E_i\cup\{\langle v_s,d_t\rangle\mid\langle v_s,v_t\rangle\in E_{G_i}^-(v_t)\}\cup\{\langle d_t,v_t\rangle\}-E_{G_i}^-(v_t)\}\\&=\{\langle v_j,v_s\rangle\mid\langle v_j,v_s\rangle\in E_i\cup\{\langle v_s,d_t\rangle\mid\langle v_s,v_t\rangle\in E_{G_i}^-(v_t)\}\cup\{\langle d_t,v_t\rangle\}\}\\&=\{\langle v_j,v_s\rangle\mid\langle v_j,v_s\rangle\in E_i\cup\{\langle v_s,d_t\rangle\mid\langle v_s,v_t\rangle\in E_{G_i}^-(v_t)\}\}\ (v_s\in SI-\{v_t\})\\&=\{\langle v_j,v_s\rangle\mid\langle v_j,v_s\rangle\in E_i\}\ (v_s\in SI-\{v_t\},v_s\neq d_t)\\&=E_{G_i}^-(v_s)\end{aligned}$$

所以,若 $v_s\in SI-\{v_t\}$,则 $E_{G_{i+1}}^-(v_s)=E_{G_i}^-(v_s)$。

假设:数据源向图 $G_{i+1}=R_{v_t}(G_i)$ 是在 G_i 上利用结点 v_t 运用引入规则操作所得到的,由性质 1 可知:在项集 SI 中,除结点 v_t 的入度变为 1 之外,其他项的入度都保持不变。

定理 1 假设在数据源向图 G_0 中存在 m 个结点 $u_1,u_2,\cdots,u_m$,且 $Deg^-(u_i)>1$,$i=1,\cdots,m$,用引入规则 $k(0\leqslant k\leqslant m)$ 次后得到数据源向图 G_k。若 $v_s\in SI-\{u_{j1},u_{j2},\cdots,u_{jk}\}$,则 $E_{G_k}^-(v_s)=E_{G_0}^-(v_s)$。

下面用数学归纳法来对运用引入规则的次数 k 进行证明。

(1)$k=0$ 时,显然,数据源向图 G_0 没有任何变化,即:$E_{G_k}^-$

$(v_s)=E^-_{G_0}(v_s)$，$\forall v_s\in SI-\{u_{j_1},u_{j_2},\cdots,u_{j_k}\}$。

(2)当 $k=t<m$ 时，若 $v_s\in SI-\{u_{j_1},u_{j_2},\cdots,u_{j_k}\}$，则 $E^-_{G_k}(v_s)=E^-_{G_0}(v_s)$。下面证明：当 $k=t+1$ 时，命题也成立。

假设：$G_{t+1}=R_{u_{j_{t+1}}}(G_t)$，$v_s\in SI-\{u_{j_1},u_{j_2},\cdots,u_{j_t},u_{j_{t+1}}\}$。

由集合论的基本知识可知：$v_s\in SI-\{u_{j_{t+1}}\}$，$v_s\in SI-\{u_{j_1},u_{j_2},\cdots,u_{j_t}\}$。

由“$v_s\in SI-\{u_{j_{t+1}}\}$”和性质 1(5)可知：$E^-_{G_{t+1}}(v_s)=E^-_{G_t}(v_s)$。

由“$v_s\in SI-\{u_{j_1},u_{j_2},\cdots,u_{j_t}\}$”和归纳假设可知：$E^-_{G_t}(v_s)=E^-_{G_0}(v_s)$。

因此，若 $v_s\in SI-\{u_{j_1},u_{j_2},\cdots,u_{j_t},u_{j_{t+1}}\}$，则 $E^-_{G_{t+1}}(v_s)=E^-_{G_0}(v_s)$。

假设数据源向图 $G_0=\langle V_0,E_0\rangle$，$V_0=SI\cup D_0$，则存在 m 个入度大于 1 的项 $u_1,u_2,\cdots,u_m$，在 G_0 的基础上运用 m 次引入规则得到图 G_m。

由引入规则可得：

$$\begin{aligned}G_m&=R_{u_m}(G_{m-1})\\&=R_{u_m}(R_{u_{m-1}}(G_{m-2}))\\&\cdots\\&=R_{u_m}(R_{u_{m-1}}(\cdots R_{u_1}(G_0)\cdots))\end{aligned}$$

3.3　引入说明的算法

由引入说明的原理可以得到如下的引入说明的算法：

算法 1　引入说明的算法

输入：n 个表$\langle T_i,SI_j\rangle$，$SI_j=\{I_{j1},I_{j2},\cdots,I_{jm}\}$，j=1，…，n，以及所有表项之间的来源关系

输出：数据源向图 G

1. 构造数据源向图 $G_0=\langle V_0,E_0\rangle$，其中：$D_0=\varnothing$，$V_0=SI\cup D_0$，$E_0=\{\langle v_i,v_j\rangle\mid v_i,v_j\in SI\}$；
2. i=0；
3. while ($\exists u_j\in SI$ and $Deg^-(u_j)>1$){
   $D_{i+1}=D_i\cup\{d_j\}$；
   $V_{i+1}=V_i\cup D_i$；
   $E_{i+1}=E_i\cup\{\langle v_s,d_j\rangle\mid\langle v_s,u_j\rangle\in E^-_{G_i}(u_j)\}\cup\{\langle d_j,u_j\rangle\}-E^-_{G_i}(u_j)$；
   $G_{i+1}=\langle V_{i+1},E_{i+1}\rangle$；
   i=i+1；
   }
4. 输出 G_{i-1}。

由引入说明的算法得到的数据源向图 $G=\langle V,E\rangle$，$V=SI\cup D$，$\forall v\in SI$，有 $Deg^-(v)\leqslant 1$。

引入说明的算法中，由性质 1(3)知，while 循环每一步操作使得 $Deg^-(u_j)=1$，且由定理 1 知该操作不会增加新的入度大于 1 的项，因此循环是可以停止的。

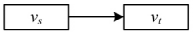
引入说明的算法的时间主要耗费在边的添加和删除操作上，添加的边集为：$ADD=\{\langle v_s,d_j\rangle\mid\langle v_s,u_j\rangle\in E^-_{G_i}(u_j)\}\cup\{\langle d_j,u_j\rangle\}$，删除的边集为：$SUB=E^-_{G_i}(u_j)$，因此一次循环共处理有向边的数量为：

$$\begin{aligned}|ADD|+|SUB|&=Deg^-(u_j)+1+Deg^-(u_j)\\&=2Deg^-(u_j)+1\leqslant 2\Delta^-+1\end{aligned}$$

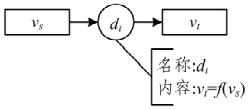
假设 SI 中有 k 个项的入度大于 1，则算法中的循环要进行 k 次，因此算法总耗时不大于 $k(2\Delta^-+1)$，所以引入说明的算法的时间复杂度为 $O(k*\Delta^-)$。

3.4　其它情况

引入说明的算法研究了数据源向图中入度大于 1 的项的处理方法。本节将研究数据源向图中入度等于 1 的项(如图 4(a)所示)的处理方法。



(a)入度为 1 的项无说明的表示方式



(b)入度为 1 的项引入说明的表示方式

图 4　数据源向图中入度等于 1 的项的表示方式

当数据源向图 $G_i=\langle V_i,E_i\rangle$ 中 $Deg^-(v_t)=1$ 且$\langle v_s,v_t\rangle\in E_i(v_s,v_t\in SI)$时，$v_s$ 和 v_t 存在如下两种关系：

(1)v_t 的值不经过任何处理直接由 v_s 得到；

(2)v_t 的值由项 v_s 通过映射 F 得到，即 $v_t=F(v_s)$。

关系(1)表示项 v_t 来源于 v_s，因此边$\langle v_s,v_t\rangle$可反映出该关系，不需要特别处理。

关系(2)表示项 v_t 来源于映射 $F(v_s)$，因此需要引入说明(如图 4(b)所示)来表示其映射关系。例如项 $v_t=3.14*v_s^2$(v_s 是项)，或者 $v_t=\Sigma_{con}Value(v_s)$(表示 v_t 的值为：将表中符合条件 con 的元组中的项 v_s 的值累加)，需要在 v_s 和 v_t 之间引入说明$\langle d,\text{“}v_t=3.14*v_s^2\text{”}\rangle$或者$\langle d,\text{“}v_t=\Sigma_{con}v_s\text{”}\rangle$来表示 v_s 和 v_t 的来源关系。在算法 1 中，由于 v_t 的入度是 1，因此无法直接引入说明。在这种情况下，需按照映射方式进行预处理。

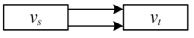
(1)存在与集合有关的映射

假设两个表$\langle T_i,SI_i\rangle$和$\langle T_j,SI_j\rangle$，若存在映射 g 使得 $v_t=g(v_s)$，$v_s\in SI_i$，$v_t\in SI_j$，若 g 的运算需要在表的元组集合中进行，则称 g 为与集合有关的映射。例如：将表中若干元组中的某项 v_s 进行求和、求最大值、求最小值等运算可以得到项 v_t，记为 $v_t=g(v_s)$。由于数据源向图中的边表示模式上的关系，在没有引入说明的情况下，映射 $v_t=g(v_s)$ 在数据源向图中表示为 v_s，v_t 之间的一条有向边$\langle v_s,v_t\rangle$(见图 4(a))。

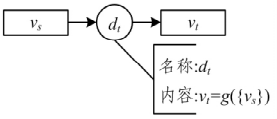
因为 $Deg^-(v_t)=1$，无法通过算法 1 引入说明，表示 v_s 和 v_t 之间的映射关系，因此这种情况无法与关系(1)进行区分。下面两种方法可在 v_s 和 v_t 之间引入说明。

1)(增加项的入度)　在数据源向图 G_0 中，若两个项 v_s 和 v_t 存在与集合有关的映射 $g:v_t=g(\{v_s\})$，则在项 v_s 和 v_t 之间建立两条平行边，如图 5(a)所示，此时，$Deg^-(v_s)=2>1$，通过算法 1 可在 v_s 和 v_t 之间增加说明(如图 5(b)所示)。

2)(直接增加说明)　在建立数据源向图项 v_s 和 v_t 之间的边时，直接引入说明$\langle d_t,\text{“}v_t=g(\{v_s\})\text{”}\rangle$，并建立连线集$\{\langle v_s,d_t\rangle,\langle d_t,v_t\rangle\}$，如图 5(b)所示。



(a)v_s，v_t 之间的平行边



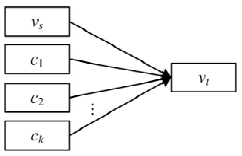
(b)引入说明后 v_s，v_t 的关系表示

图 5　数据源向图中 $v_j=g(\{v_i\})$ 的处理方式

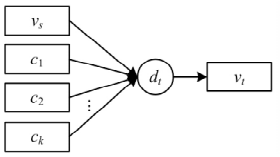
(2)与集合无关的映射

假设有两个表$\langle T_i, SI_i\rangle$和$\langle T_j, SI_j\rangle$,若存在映射 f 使得 $v_t=f(v_s)$,$v_s\in SI_i$,$v_t\in SI_j$,f 不是与集合有关的映射,f 中包含 k 个常数 $c_1, c_2, \cdots, c_k$,且称 f 为与集合无关的映射。v_s 和 v_t 的关系在未引入说明的 G_0 中的表现形式是:v_s,v_t 间存在有向边$\langle v_s, v_t\rangle$。如图 4(a)所示,$Deg^-(v_t)=1$,与情况 1 类似,也通过两种方法引入说明。

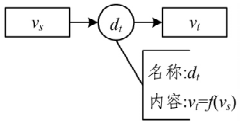
1)(增加项的入度) 若表$\langle T_i, SI_i\rangle$和$\langle T_j, SI_j\rangle$中存在与集合无关的映射 $f: v_t=f(v_s)$,$v_s\in SI_i$,$v_t\in SI_j$,其中 f 包含 k 个常数 $c_1, c_2, \cdots, c_k$,则在建立数据源向图时,应增加独立的项表示常数 $c_1, c_2, \cdots, c_k$,并建立连线集$\{\langle v_s, v_t\rangle, \langle c_1, v_t\rangle, \langle c_2, v_t\rangle, \cdots, \langle c_k, v_t\rangle\}$(如图 6(a)所示)。此时,$Deg^-(v_t)=k+1>1$,运用引入说明的算法处理后(如图 6(b)所示),再将孤立的常数项从数据源向图中删除(如图 6(c)所示),即可在项 v_s 和 v_t 之间引入说明。



(a)引入独立的项表示常数



(b)通过算法 1 引入说明 d_t



(c)引入说明后 v_s,v_t 关系的最终表示

图 6 数据源向图中项入度为 1 且来源于非集合运算的映射的处理

2)(直接增加说明) 在建立数据源向图项 v_s 和 v_t 之间的边时,直接引入说明$\langle d_t$,"$v_t=f(v_s)$"$\rangle$并建立连线集$\{\langle v_s, d_t\rangle, \langle d_t, v_t\rangle\}$,如图 5(c)所示。

由 2.1—2.3 节及本节可知:对于数据源向图中入度大于 1 或等于 1 的项,都可运用引入说明的算法进行处理。

在实际项目的设计中,一般采用直接增加说明的方法来引入说明,实际上本文的算法 1 就描述了数据源向图建立的整个思维过程:发现有关系的几个字段,再通过说明体现出它们之间的关系。

4 应用实例

本节主要以某出版社成本管理系统作为实例来展示数据源向图在实际工程中的使用情况,以说明数据源向图在实际工程中的使用方式以及对数据项之间关系的表示情况。本系统涉及到的表较多,实际设计中为了便于展示,一个设计可能会有多个子图。下文为了简化展示,都选择其中重要的一个部分进行展示。

该软件主要帮助出版社完成编辑的工作量汇总及费用摊分等功能。本系统已经正常使用超过 5 年,目前依然在使用中。图 7 为该系统的数据设计对应的数据源向图。

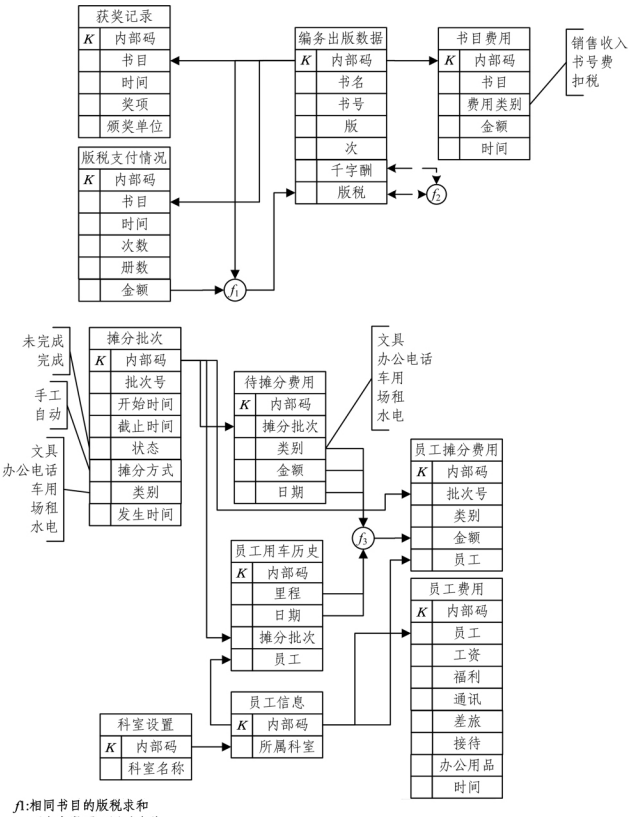


图 7 某出版社成本管理系统

从图 7 中可以看出每个表之间的数据流向,例如书目信息统一来自于"编务出版数据"表的关键字"内部码";成本摊分的批次统一来自于"摊分批次"表等。还可以通过图 7 了解到一些字段的计算方法,例如:"编务出版数据"表中的〈版税〉字段是通过"版税支付情况"表按照〈书目〉将〈金额〉累加获得的;"员工摊分费用"表中的〈金额〉字段则是"待摊分费用"表中按照不同〈类别〉通过不同的方式计算得到的。该系统在设计的同时对数据源向图进行了一些补充,例如:为了更清晰地表示数据之间的约束关系,用说明 f_2 对"编舞出版数据"表中互斥的两个项(〈版税〉和〈千字酬〉不可同时存在)进行标记,并用虚线区分一般的说明。

结束语 本文主要通过实践提出了一种新的数据库设计中的数据表示工具:数据源向图,并对其进行了形式化定义,说明了如何通过数据源向图清晰、详细地体现出数据库中项之间的关系;本文还介绍了通过数据源向图从需求形成设计的过程,并给出了实际应用中的案例。

有了形式化定义后,可以针对数据库设计进行其他程序化工作,例如通过研究数据源向图节点和连线之间的关系,形式化地进行数据库设计中冗余的消除工作等。

同时由于业务系统的业务流程是通过数据流动体现出来的,而数据源向图中项之间的连线正好体现出了数据流动的方向,因此数据源向图不仅可以作为数据库设计的工具,也应该可以作为系统整体设计的工具。下一步计划将数据源向图更深入地引入到数据库设计中来。

(下转第 509 页)

长度增加而显著下降。进而可以得出，随着短数据包长度的增加，数据包的传输延迟也显著增加，总线性能显著下降。

4）总线性能随着在总线上传输长数据包的节点数量占总节点数比例的减小而提升。从图 9 可以得出，在其他参数不变和长数据包数量一定的情况下，当从节点数量增加时数据包的传输延迟减小。

5）该模型在理论上与 UM-BUS 的特性十分匹配，在未来的工作中将在 UM-BUS 实际的应用平台上对其有效性进行进一步的验证。

结束语　UM-BUS 总线是一种具有动态容错能力和远程穿透能力的高速串行总线。基于排队论的理论提出了针对 UM-BUS 总线的性能建模与评估方法，该方法定性地描述了数据包在总线上的传输特征，定量地分析了数据包在总线上传输的最大延迟、最小延迟以及平均延迟。文中的工作为总线在实际系统中的应用提供了可靠的理论支持。

参 考 文 献

[1] 王嘉佳. 动态可重构总线控制器的设计与实现[D]. 北京：首都师范大学，2012.

[2] 张伟功，丁瑞，关永，等. 一种可动态重构的高速串行总线系统及控制方法：ZL200910180480. 3[P]. 2009.

[3] 王行仁. 建模与仿真技术的发展和应用[J]. 机械制造与自动化，2010，39(1)：16-45.

[4] 胡玥. 基于 ICD 的小卫星平台电子学仿真测试系统研究[D]. 北京：中国科学院空间科学与应用研究中心，2006.

[5] 常同立. 空间对接动力学半物理仿真系统设计及试验研究[D]. 哈尔滨：哈尔滨工业大学，2007.

[6] 刘延斌，金光. 半实物仿真技术的发展现状[J]. 光机电信息，2003(1)：27-32.

[7] 李荣民，张朝明，王锴，等. 嵌入式系统和 CAN 总线技术的优越性以及应用[J]. 现代工业经济和信息化，2013(2)：66-68.

[8] 丁楠. 基于 PCI 总线的非标准并行通信总线仿真接口的研制[D]. 西安：西北工业大学，2007.

[9] TI. MLVDS Signaling Rate Versus[R]. TI Application Report SLLA127. 2003.

[10] CENA G. Evaluation of Ether CAT Distributed Clock Performance[J]. Industrial Informatics，2012，8(1)：20-29.

[11] DURANTE L，VALENZANO A. On the performance of the IEC61158fieldbus[J]. Computer Standards & Interfaces，1999，21(3)：241-250.

[12] HONG S H，KO S J. Analysis of real time data transmission in the DLLof IEC/ISA field bus[C]// Proceeding of ISIE' 98. 1998：694-699.

[13] RAHMAN M A，ANWAR F，NAEEM J，et al. A simulation based performance comparison of routing protocol on Mobile Adhoc Network (proactive，reactive and hybrid)[C]// 2010 International Conference on Computer and Communication Engineering(ICCCE). 2010：1-5.

[14] KLAR R A，BERTRAND A R. The Evolution of Space Wire：A Comparison to Established and Emerging Technologies[C]// The 3th International Space Wire Conference. 2010.

[15] OTHMAN H F，AJI Y R，FAKREDDIN F T，et al. Controller Area Networks：Volution and Applications[C]// 2nd Information and Communication Technologies. 2006：3088-3093.

[16] BLANC J P C. On the numerical inversion of busy period related transforms[J]. Operations Research Letters，2002，30(1)：33-42.

（上接第 474 页）

参 考 文 献

[1] LEI J，THODOROS T，ALEX B，et al. Goal-Oriented Conceptual Database Design[C]// 15th IEEE International Requirements Engineering Conference. 2007.

[2] SUSANNE P，Evolution of Entity-Relationship Modelling[J]. Data & Knowledge Engineering，2006，56(2)：122-138.

[3] CHEN P P. The entity-relationship model：Toward a unified view of data[J]. ACM Trans. Database Syst. ，1976，1(1)：9-36.

[4] 陈冰川，吴向军，王和勇，等，基于数据源向图的数据项的表示与获取方法[J]，电子学报，2012，40(11)：2239-2246.

[5] CODD E F. Further normalization of the data base relational model[M]// Rustin R，ed. Courant Institute Computer Science Symposia Series：Data Base Systems，Prentice-Hall，Englewood Cliffs，NJ，1972.

[6] CHEN P P. The entity-relationship model：Toward a unified view of data[J]. ACM Trans. Database Syst. 1976，1(1)：9-36.

[7] TEORY T J. A Logical Design Methodology for Relational Databases Using the Extended Entity-Relationship Model[J]. Computing Surveys，1986，18(2)：197-222.

[8] DEY D，STOREY V C，BARRON T M. Improving Database Design Through the Analysis of Relationships[J]. ACM Transactions on Database Systems，1999，24(4)：453-486.

[9] NOAH S A M，WILLAMS M. Intelligent Object Analyser For Conceptual Database Design Model[J]. Jurnal Teknologi，2003，39(D)：27-44.

[10] STOREY V C，GOLDENSTEIN R C，ULLRICH H. Naive Semantics to Support Automated Database Design[J]. IEEE Transaction on Knowledge and Data Engineering，2002，14(1)：1-12.

[11] LLOYD-WILLIAMS M，BEYNON-DAVIES P. Expert system for database design：a comparative review[J]. Artificial Intelligence Review，1992，6(3)：263-283.

[12] 杨冬青，唐世渭. 数据库设计工具集 DBTOOLS 的设计[J]. 软件学报，1993，4(4)：26-31.

[13] SONG I Y，KHARE R，DAI B. SAMSTAR：A Semi-Automated Lexical Method for Generating Star Schemas from an Entity-Relationship Diagram[C]// DOLAP 07. Lisbon，Portugal，November 2007.

[14] LEI J，THODOROS T，ALEX B，et al. Incorporating Goal Analysis in Database Design：A Case Study from Biological Data Management[C]// the 14th IEEE International Requirements Engineering Conference. IEEE Computer Society，2006：196-204.