

基于数据源向图的数据项的表示与获取方法

陈冰川¹, 吴向军², 王和勇³, 李 磊¹

(1. 中山大学信息科学与技术学院软件研究所, 广东广州 510275;

2. 中山大学软件学院, 广东广州 510275; 3. 华南理工大学电子商务学院, 广东广州 510641)

摘 要: 如何清晰的表示数据项之间的关系, 进而了解数据之间更深层的内在关系, 在当前的需求分析中没有明确的方法, 但是其对于软件需求中数据的获取和数据关系的表示具有重要作用. 如何通过数据本身采用推导的方法获取软件系统所必须的数据以及数据项, 在当前的需求分析和获取方法中也没有直接方法和思路. 本文采用数据源向图的方法, 通过推导的方法获取必要的数据和数据项, 并通过它来表示数据中各个数据项之间的关系, 使得数据项之间关系的表示更加清晰, 数据项的获取更加容易.

关键词: 需求获取; 数据源向图; 不动点; 数据字典; 需求分析

中图分类号: TP302.7 **文献标识码:** A **文章编号:** 0372-2112 (2012) 11-2239-08

电子学报 URL: <http://www.ejournal.org.cn> **DOI:** 10.3969/j.issn.0372-2112.2012.11.016

The Method of Data Item Indicating and Eliciting Based on DFDC

CHEN Bing-chuan¹, WU Xiang-jun², WANG He-yong³, LI Lei¹

(1. Software Research Institute in College of Information Science and Technology, Sun Yat-Sun University, Guangzhou, Guangdong 510275, China;

2. School of Software, Sun Yat-Sun University, Guangzhou, Guangdong 510275, China;

3. E-Commerce School, South China University of Technology, Guangzhou, Guangdong 510641, China)

Abstract: How to unambiguously represent the relationship between data items and then extract the deeper level relationship between them is playing an important role in requirement engineering, but good methods for it doesn't exist in current requirement analysis methods. At present there also doesn't have method or thoughts for how to extract the data and data item of software system from data of requirement by deduce. In this paper, we use the method of Data Floating Direction Chart (DFDC) to extract the necessary data and data item for software systems and to represent the relationship between data items. The DFDC can make representation of relationship between data items more clearly and extraction process of data items more easily.

Key words: requirement eliciting; data floating direction chart (DFDC); fixed point; data dictionary; requirement analysis

1 引言

如何清晰的表示数据项之间的关系, 进而了解数据之间更深层的内在关系, 在当前的需求分析中没有明确的方法, 但是其对于软件需求中数据的获取和数据关系的表示具有重要作用. 如何通过数据本身采用推导的方法获取软件系统所必须的数据以及数据项, 在当前的需求分析和获取方法中也没有直接方法和思路.

当前针对数据项之间关系的研究, 除了依赖关系 (Dependant Relationship) 外, 其它关系的研究较少. 依赖关系主要是讨论单个业务表中项之间的依存关系, 通过依赖关系, 使设计者程序化地将数据库中的表进行拆分, 使其符合指定的范式例如: 3NF.

然而实际工程中发现, 获取不同业务表单之间的逻辑关系 (指数据项取值的来源关系) 并清楚地表示出来, 对需求分析和获取, 乃至设计过程都具有指导意义.

例如, 在生产系统中, 用户要求系统能够得到每个产品的生产日报: 生产日报 (日期, 产品名称, 产品型号, 产量, 次品量, 次品率). 如图 1 所示.

通过用户给出的输出结果如产品日报, 如果从数据

生产日报DailyReport
日期Date
产品名称ProductName
产品型号ProductModel
产量Yield
次品量DefectiveAmount
次品率DefectiveRate

图1 生产日报表

项的逻辑关系入手,需求获取工作将会相对容易,因为可以逐个对字段的来源进行追溯:如图 2 所示,日期根据当前时间获得、产品名称和产品型号的来源为企业的《产品目录》、产量通过《班组报产表》统计获得、次品量通过质检部门的《质检日报》获得、次品率通过产量和次品量计算获得;由此可知需要进一步获取如下业务表单《产品目录》、《班组报产表》和《质检日报》,再对这些表单重复进行上述过程,就可以针对《生产日报》获取与其相关的全部业务表单与字段。

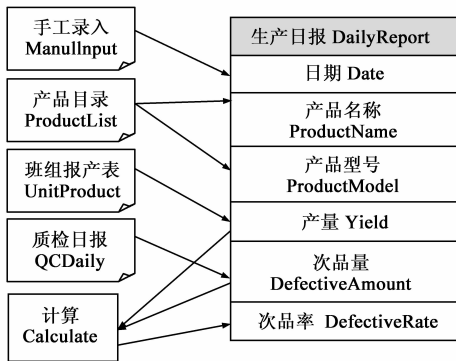


图2 按字段追溯项取值的来源

上述的迭代过程,可以获取输出表单的所有业务表单和字段,并反映出各数据项之间的逻辑关系,本文基于该迭代的方法,采用数据源向图^[1]来获取必要的业务表单和数据项,并通过它来表示数据中各个数据项之间的关系。

2 相关工作

需求工程的活动主要分为五个阶段^[2],分别是:需求获取、需求建模、形成需求规格、需求验证和需求管理。本文的方法将在需求获取和需求建模(分析)两个阶段进行讨论。

2.1 需求获取方法

需求获取是需求分析的基础,目前需求获取方法包括:面向目标的方法^[3]、面向方面(Asspect-Oriented)的需求方法^[4,5]、基于环境建模方法^[6]、基于情景的方法^[7]、为了获取需求变化的 MOPISM^[8]方法等。

陈小红等在文献^[9]中结合基于环境建模方法和基于情景的方法,提出了从交互情景出发的需求获取方法。该方法通过工具获取用户需求,最终生成需求描述文档(情景文档)。该方法给出了需求获取的规范化的过程和工具。

上述多种需求获取的方法,主要是通过访谈、问卷以及需求者经验获取软件的功能性需求,对于数据的获取缺乏足够的重视。对于如何获取与需求相关的数据以及判断数据的完整性缺少相应的描述。

2.2 需求分析方法

需求分析方法大致分为四类^[2]:面向过程的分析方法、面向数据的方法、面向控制的方法和面向对象的方法。

面向过程的分析方法以传统的结构分析方法以 Structure Architecture(SA)^[10]为代表,该类方法主要研究系统的输入输出转化的方式,对于数据本身并不是很重视^[2]。

面向数据的方法以 JSD^[11]方法为代表,该方法主要通过建立 Jackson 图来表示数据间的顺序结构、选择结构、重复结构和可选结构,反应出数据之间的构成关系,但是对于数据项之间的关系并未涉及。

面向控制的方法包括了以数据流图为主的方法,该类方法主要通过数据流图来表示数据在系统中的流向,但并未表示出数据项之间的关系。

面向对象的方法以 OOA^[12]为代表,该类方法主要通过统一建模语言 UML(Unified Modeling Language)进行建模,通过类图、对象图和包图表示出数据的载体——类和对象之间的关系。目前面向对象的方法已经成为需求分析的主要方法,因此在此基础上出现了众多基于面向对象的需求分析方法,如 Booch^[13]方法、OOSE^[14]方法以及基于类图语义框架的中文需求分析方法^[15]等,无论早期的还是改进后的面向对象的方法,都对于属性之间除了依赖关系之外其他关系未进行更多描述。

随着针对需求分析的不断深入研究出现了更多的方法:面向方面(Asspect-Oriented)的需求分析方法^[5]、基于责备(Blamed-Based)的需求不一致的分析方法^[16]、基于问题框架方法的本体制导方法^[17]等。

除此之外郑云祥等在文^[18]中提出了基于主谓宾逻辑的需求方法,该方法以宾语(业务表单)为核心进行需求分析,通过表单流程图描述了数据之间的关系,充分体现出数据在需求中的重要性。该方法虽然清楚地描述了数据之间的转换关系,但是对于如何转换以及数据项之间的关系如何表现并未给出详细论述。

本文针对当前需求获取和分析方法存在的问题,提出了基于数据源向图的方法,解决如何获取针对输出要求的完整的数据项,以及如何表示数据项之间关系问题,并以此作为需求获取和需求分析的必要补充。

3 数据源向图

本文提出了一种新的数据项关系表示形式——数据源向图 DFDC(Data Floating Direction Chart),它是一种有向图,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。

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 $\langle \text{iname}, f \rangle$, iname 是项的名称, $f(\{K, \text{NULL}\})$ 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 $\langle \text{tname}, SI \rangle$, tname 是表名, $SI = \{I_1, I_2, \dots, I_n\}$ 是项的集合 (Set of Item), $I_j = \langle \text{iname}_j, f_j \rangle, j = 1 \dots n$ 。

为方便文中叙述, 本文用 iname 表示项 $\langle \text{iname}, f \rangle$, 用 tname 表示表 $\langle \text{tname}, SI \rangle$, 用符号 $T. I_j$ 表示表 T 的第 j 个项, $\text{GetTable}(I)$ 表示项 I 所在的表, 如: $\text{GetTable}(T. I_j) = T$ 。

下面用数据源向图 DFDC 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 $\langle T_i, SI_j \rangle, SI_j = \{I_{j1}, I_{j2}, \dots, I_{jm}\}, j = 1 \dots n$ 。其数据源向图 $G_0 = \langle V_0, E_0 \rangle$ 为有向图,

$$(1) V_0 = SI = \bigcup_{j=1 \dots m} SI_j.$$

(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。

假设有三个表 $\langle T_1, SI_1 \rangle, \langle T_2, SI_2 \rangle$ 和 $\langle T_3, SI_3 \rangle, SI_1 = \{ \langle \text{Item}_{11}, K \rangle, \dots, \langle \text{Item}_{1s}, \text{NULL} \rangle, \dots \}, SI_2 = \{ \langle \text{Item}_{21}, K \rangle, \dots, \langle \text{Item}_{2t}, \text{NULL} \rangle, \dots \}, SI_3 = \{ \langle \text{Item}_{31}, K \rangle, \dots, \langle \text{Item}_{3w}, \text{NULL} \rangle, \dots \}$ 。

在这三个表中, 其项之间的部分关系为: $\text{Item}_{32} = \text{Item}_{11}, \text{Item}_{3w} = \text{Item}_{1s} + \text{Item}_{2t}$ 。

由定义 3 可知: $V_0 = SI_1 \cup SI_2 \cup SI_3, E = \{ \langle \text{Item}_{11}, \text{Item}_{32} \rangle, \langle \text{Item}_{1s}, \text{Item}_{3w} \rangle, \langle \text{Item}_{2t}, \text{Item}_{3w} \rangle \}$ 。

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。

由图 3 可知: 项 $T_3. \text{Item}_{32}$ 的值来源于项 $T_1. \text{Item}_{11}$; 项 $T_3. \text{Item}_{3w}$ 值的来源与项 $T_1. \text{Item}_{1s}$ 和 $T_2. \text{Item}_{2t}$ 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 $\langle \text{dname}, \text{dsc} \rangle$, dname 是说明的名称 (或标记), dsc 是说明的内容。

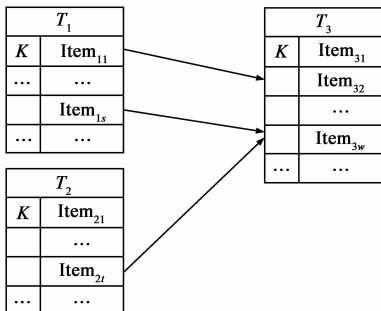


图3 T_1, T_2 和 T_3 之间部分关系的数据源向图 G_0

为方便叙述, 用 dname 表示说明 $\langle \text{dname}, \text{dsc} \rangle$, 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 形式。

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

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

$$(2) E = E^0 \cup E^1 \cup E^2, \text{ 其中:}$$

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

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

$$(c) E^2 = \{ \langle d_i, v_s \rangle \mid v_s \in SI, d_i \in D \}, \forall \langle d_i, v_s \rangle \in E^2 \text{ 表示 } v_s \text{ 的值来源于说明 } d_i.$$

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

$$(1) V_0 = SI \cup D, \text{ 说明集 } D = \emptyset.$$

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

由于本文的研究以数据源向图为基础, 为此基于图论的概念中给出下列定义:

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

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

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

$$(3) \text{ 图 } G \text{ 中最大入度 } \Delta^- \text{ 表示 } \text{Max}(\{ \text{Deg}^-(v_i), v_i \in S \}).$$

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

3.2 引入说明的原理

在数据源向图 G 中, 若 $v_i \in SI, \text{Deg}^-(v_j) > 1$ (如图 4(a)), 则结点 v_j 值的获得与结点 v_1, v_2, \dots, v_k 的值相关。这时, 引入说明 $\langle d_j, \text{“结点 } v_j \text{ 值的获得来源于结点 } v_1, v_2, \dots, v_k \text{ 的值”} \rangle$, 并通过“引入说明的规则”进行操作, 使之变为如图 4(b)所示的关系。

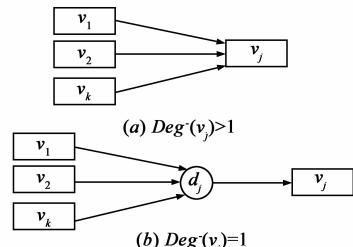


图4 引入说明的变化示意图

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

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

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

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

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

对图 3 所示的数据源向图 G_0 利用引入说明的规则后, 可得到的数据源向图 G_1 如图 5 所示.

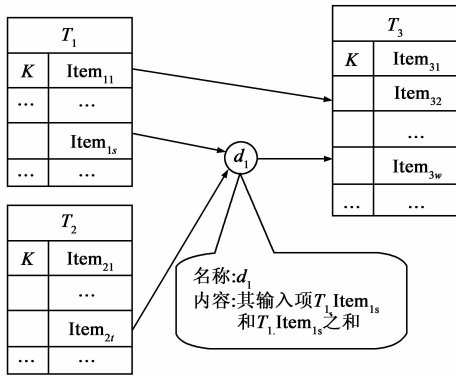


图5 数据源向图 G_1

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

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

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

(2) $E_{i+1} = E_i \cup \{ \langle v_s, d_i \rangle | \langle v_s, v_i \rangle \in E_{G_i}^-(v_i) \} \cup \{ \langle d_i, v_i \rangle \} - E_{G_i}^-(v_i)$.

(3) $Deg^-(v_i) = 1$.

(4) $Deg^-(d_i) = k, Deg^+(d_i) = 1$.

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

证明

(1) 性质 1(1) ~ (3) 可直接由引入规则得到.

(2) 性质 1(4).

由引入规则可知: 以 d_i 作为终点的边集为 $\{ \langle v_s, d_i \rangle | \langle v_s, v_i \rangle \in E_{G_i}^-(v_i) \}$, 以 d_i 作为始点的边集为 $\{ \langle d_i, v_i \rangle \}$,

$d_i, v_i \rangle \}$,

所以, $Deg^-(d_i) = k, Deg^+(d_i) = 1$.

(3) 性质 1(5)

假设: $v_s \in SI - \{v_i\}$.

由于数据源向图 G_{i+1} 是对图 G_i 利用结点 v_i 运用引入规则所得到的, 所以, 其边集的变化为:

$$E_{i+1} = E_i \cup \{ \langle v_s, d_i \rangle | \langle v_s, v_i \rangle \in E_{G_i}^-(v_i) \} \cup \{ \langle d_i, v_i \rangle \} - E_{G_i}^-(v_i)$$

由定义 6 可知:

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

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

假设: 数据源向图 $G_{i+1} = R_{v_i}(G_i)$ 是在 G_i 上利用结点 v_i 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_i 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, \dots, u_m , 且 $Deg^-(u_i) > 1, i = 1 \dots m$, 用引入规则 k ($0 \leq k \leq m$) 次后得到的数据源向图 G_k . 若 $v_s \in SI - \{u_{j_1}, u_{j_2}, \dots, u_{j_k}\}$, 则, $E_{G_k}^-(v_s) = E_{G_0}^-(v_s)$.

证明

下面对运用引入规则次数 k 用数学归纳法来证明.

(1) $k = 0$ 时

显然, 数据源向图 G_0 没有任何变化, 即: $E_{G_k}^-(v_s) = E_{G_0}^-(v_s), \forall v_s \in SI - \{u_{j_1}, u_{j_2}, \dots, u_{j_k}\}$.

(2) 当 $k = t < m$ 时, 若 $v_s \in SI - \{u_{j_1}, u_{j_2}, \dots, u_{j_k}\}$, 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: $G_{t+1} = R_{u_{j_{t+1}}}(G_t), v_s \in SI - \{u_{j_1}, u_{j_2}, \dots, u_{j_t}, u_{j_{t+1}}\}$.

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

$$- \{u_{j_1}, u_{j_2}, \dots, u_{j_i}\}.$$

由“ $v_s \in SI - \{u_{j_{i+1}}\}$ ”和性质 1(5)可知: $E_{G_{i+1}}^-(v_s) = E_{G_i}^-(v_s)$.

由“ $v_s \in SI - \{u_{j_1}, u_{j_2}, \dots, u_{j_i}\}$ ”和归纳假设可知: $E_{G_i}^-(v_s) = E_{G_0}^-(v_s)$.

所以,若 $v_s \in SI - \{u_{j_1}, u_{j_2}, \dots, u_{j_i}, u_{j_{i+1}}\}$, 则,

$$E_{G_{i+1}}^-(v_s) = E_{G_0}^-(v_s).$$

假设数据源向图 $G_0 = \langle V_0, E_0 \rangle$, $V_0 = SI \cup D_0$, 存在 m 个入度大于 1 的项 u_1, u_2, \dots, 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:

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

3.3 引入说明的算法

算法 1 引入说明的算法

输入 n 个表 $\langle T_i, SI_j \rangle$, $SI_j = \{I_{j_1}, I_{j_2}, \dots, I_{j_m}\}$, $j =$

$1..n$, 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: $D_0 = \emptyset$, $V_0 = SI \cup D_0$, $E_0 = \{ \langle v_i, v_j \rangle \mid v_i, v_j \in SI \}$;

(2) $i = 0$;

(3) while ($\exists u_j \in SI$ and $Deg^-(u_j) > 1$) { // 依次处理入度大于 1 的项

$D_{i+1} = D_i \cup \{d_j\}$; // 加入说明

$V_{i+1} = V_i \cup D_i$; // 将说明作为节点

$E_{i+1} = E_i \cup \{ \langle v_s, d_j \rangle \mid \langle v_s, u_j \rangle \in E_{G_i}^-(u_j) \} \cup \{ \langle d_j, u_j \rangle \} - E_{G_i}^-(u_j)$; // 将 u_j 原来的入边指向说明, 再将说明和 u_j 之间增加一条指向 u_j 的有向边.

$G_{i+1} = \langle V_{i+1}, E_{i+1} \rangle$;

$i = i + 1$; }

(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 $Deg^-(v) \leq 1$.

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耗费主要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:

$$|ADD| + |SUB| = Deg^-(u_j) + 1 + Deg^-(u_j)$$

$$= 2Deg^-(u_j) + 1$$

$$\leq 2\Delta^- + 1$$

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为 $k(2\Delta^- + 1)$, 所以引入说明的算法的时间复杂度为 $O(k * \Delta^-)$.

3.4 小结

通过数据源向图的定义和说明的引入规则可以看出, 通过数据源向图, 不仅能够展示数据项之间简单的来源关系, 还能够表示数据项之间的来源运算关系.

4 通过数据源向图获取数据项

数据字典的获取是需求分析中的一项重要工作, 传统做法就是简单的将需求中搜集到的业务表单中的数据项简单罗列, 但是这种方法无法分辨获取的数据是否足够. 而采用数据源向图的方法可以判断出获取的数据字典, 所有的数据项是否完全.

基于数据源向图获取数据字典的基本思路是:

(1) 将用户要求的必需输出的表加入到数据源向图中.

(2) 在数据源向图中找到项 v 其 $Deg^-(v) = 0$.

(3) 判断 v 的值是否需要引用其他表单的数据, 或参照其他表单数据的计算结果.

(4) 如果需要, 则在数据源向图中加入所需的表和相关的项.

(5) 反复操作直到所有的入度为 0 的项都无法引入新表为止.

为了说明基于数据源向图的数据获取方法, 首先要对数据项进行分类.

4.1 数据项的分类

根据表中数据项来源的不同, 可以将数据项分为两类.

定义 7 基础项 BI(Base Item): 假设数据库表或者业务表中的项 v_i , 如果 v_i 在实际工作中或者在当前系统范围内, 都是通过手工录入而不需要参照其它业务表单时, 称 v_i 是基础项.

例如, 图 1 中日期是基础项.

定义 8 导出项 DI(Derivation Item): 假设数据库表或者业务表中的项 v_i 不是基础项, 则 v_i 是导出项.

例如, 图 1 中产品名称, 产品型号, 产量, 次品量, 次品率是导出项.

4.2 数据项的获取

定义 9 设 V 为项的集合, 定义 $GS(V)$ 为 V 中所有导出项的来源项所在表的项的集合.

由定义 9 可知:

(1) $GS(V)$ 是一个项的集合;

(2) $GS(V)$ 的每个表的项中都能找到一个项是 V 中某个导出项的来源即: $\forall SI_{T_i} \subseteq GS(V) \exists v' \in SI_{T_i}$, 使得 $\exists v \in V$ 有 v 来源于 v' 或来源于 v' 与其它项运算结果;

(3) V 中的任何一个导出项都可以在 $GS(V)$ 的某个表中找到其来源项即: $\forall v \in V$ 且 v 是导出项则 $\exists SI_{T_i} \subseteq GS(V)$, $\exists v' \in SI_{T_i}$, 使得 v 来源于 v' 或来源于 v' 与其它项的组合;

(4) 如果 V 中的项都是基础项则 $GS(V) = \emptyset$;

(5) $GS(V_1 \cup V_2) = GS(V_1) \cup GS(V_2)$.

例如 若业务表单 T_1, T_2, T_3 具有图 3 中的来源关系, 则 $GS(SI_{T_3}) = SI_{T_1} \cup SI_{T_2}$, $GS(\{Item_{3w}\}) = SI_{T_1} \cup SI_{T_2}$, $GS(\{Item_{32}\}) = SI_{T_1}$.

定理 2 若 VI 是所有业务表单项的幂集 (Power Set), 若定义 VI 上的映射 $GetTableItem(X) = GS(X) \cup X$, $X \in VI$ 则 $GetTableItem(X)$ 具有不动点.

证明

(1) 因为 VI 是幂集所以 $\langle VI, \subseteq \rangle$ 是完全格

(2) 若 $V_1 \in VI, V_2 \in VI$ 且 $V_1 \subseteq V_2$, 不妨设 $V_2 = V_1 \cup V'$ 则

$$\begin{aligned} GetTableItem(V_2) &= GS(V_2) \cup V_2 \\ &= GS(V_1) \cup GS(V') \cup V_1 \cup V' \\ &= GetTableItem(V_1) \cup GetTableItem(V'), \end{aligned}$$

即 $GetTableItem(V_1) \subseteq GetTableItem(V_2)$, 因此映射 $GetTableItem(X)$ 是单调的.

(3) 由(1)和(2)根据不动点定理可知在 VI 上的映射 $GetTableItem(X)$ 具有不动点.

令 $GetTableItem(GetTableItem^{n-1}(X)) = GetTableItem^n(X)$ 并且 $GetTableItem(X) = GetTableItem^1(X)$.

若 V 是输出表单项的集合, K 是 $GetTableItem(V)$ 的不动点, 则由映射 $GetTableItem(X)$ 的定义可以知道 $GetTableItem^K(V)$ 记录了所有包括表单 V 在内的所有与 V 相关的业务表单项的集合.

因此以输出表项的集合 V 作为输入, 通过该映射对应的不动点算法可以获得与 V 相关的所有业务表单及项的集合.

算法 2 AllTable(V) (获取与 V 相关的所有表单及项)

输入 输出表项的集合 V

输出 与 V 相关的所有表项的集合

(1) $New = GetTableItem(V)$;

(2) $Old = V$;

(3) While ($New \neq Old$) {

(a) $Old = New$;

(b) $New = GetTableItem(New)$;

(4) Return New ;

因为映射 $GetTableItem(X)$ 有不动点, 所以算法 2 能够停机.

若已知 n 个输出报表, 反复运用算法 2, 则可以得到要求报表所有必需的业务表单项的集合 SI . 根据数据源向图是完整的定义可知, 以 SI 建立起来的数据源向图是完整的.

为了表示在数据源向图 $G = \langle V, E \rangle$ 的基础上建立 $GetTableItem(SI_{T_i})$, 为了便于描述将数据源向图 G 作为参数传入, 使映射转化为 $GetTableItem(SI_{T_i}, G)$, 其算法如下:

算法 3 $GetTableItem(OV, G)$

输入 项集合 $OV, G = \langle V, E \rangle$

输出 项集合 OV 和相应的数据源向图 G

(1) $V = V \cup OV$;

(2) For each S in $GS(OV)$

(a) For each v in S

i. $V = V \cup \{v\}$;

ii. $E = E \cup \{ \langle v, v' \rangle \mid v' \text{ 来源于 } v, v \in S, v' \in OV \}$;

(3) $T = GS(OV) \cup OV$;

(4) $OV = T$;

(5) Return OV, G ;

用经过重载后的 $GetTableItem(OV, G)$, 对算法 2 进行修改:

算法 4 CRATEDFDC (SI, G)

输入 输出表项的集合 $SI, G = \langle V, E \rangle$

输出 针对 SI 完整的数据源向图 G

(1) $Old = SI$;

(2) $New = GetTableItem(SI, G)$;

(3) While ($New \neq Old$) {

(a) $Old = New$;

(b) $New = GetTableItem(New, G)$;

(4) Return G ;

同样的若已知 n 个输出报表, 若在算法 4 开始前令 $V = \emptyset, E = \emptyset$, 反复运用 k 次算法 4 后, 再应用算法 1 引入说明, 则可以得到要求报表所有必需的业务表单项组成的数据源向图.

因此以目标表单出发, 建立数据源向图的过程, 也可以找到与目标表单相关的所有的业务表单和项.

5 实例

本章将根据一个简单的实例说明上述算法在数据项获取以及数据项关系表示中的效果. 假设用户给出了生产日报为系统的必要输出报表. 根据算法 4 可以得到如图 6 所示的数据源向图 G_0 .

可以看出生产日报中的项产量、次品量、次品率的入度大于1,因此在运用引入说明的算法后,需要将这些项和与其相关的来源项之间假如说明.假设算法 1 在图 6 为从左向右从上到下获取入度大于 1 的项,于是经过算法 1 的第一轮循环得到 G_1 如图 7 所示.

经过算法 1 三轮循环后得到数据源向图 G_3 ,如图 8 所示.

通过上述例子可以看出,根据用户给定的生产日报,采用一边建立数据源向图一边获取其必须的业务报表,最终不仅获取了所有与生产日报有关的业务表单,而且还清晰的表示了数据项之间的关系.

6 结论及以后工作

本文主要通过形式化定义数据源向图,说明通过数据源向图如何清晰的表示数据项之间的关系,然后介绍如何通过数据源向图根据要求输出的报表,找到所有必要的表和项,使数据获取过程有了依据,避免了完全依靠客户提供的被动局面,使需求过程中数据字典的获取是否完整有了评判标准.

根据数据源向图的有向图特性,以及通过溯因的数据项获取方法可知,根据用户给定的数据需求建立数据源向图,通过图是连通以及是否包含最终的输出数据来删减与输出要求无关的业务表单,从而保证了获取到的数据为最小且完整的.因此如何根据用户给定的需求,获取最小且完整的需求是下一步的重点工作.由于从数据库的视角来看,实际业务系统的流程是数据的转存过程,而数据源向图中项之间的连线正是体现出了数据流动的方向,因此数据源向图在系统整体设计中也能够起到辅助作用,因此下一步的研究方向

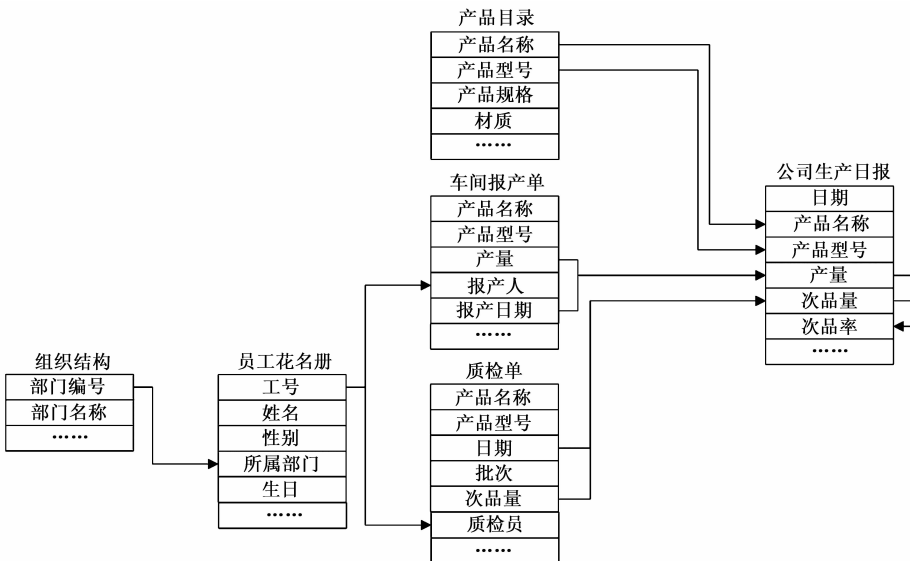


图6 数据源向图 G_0

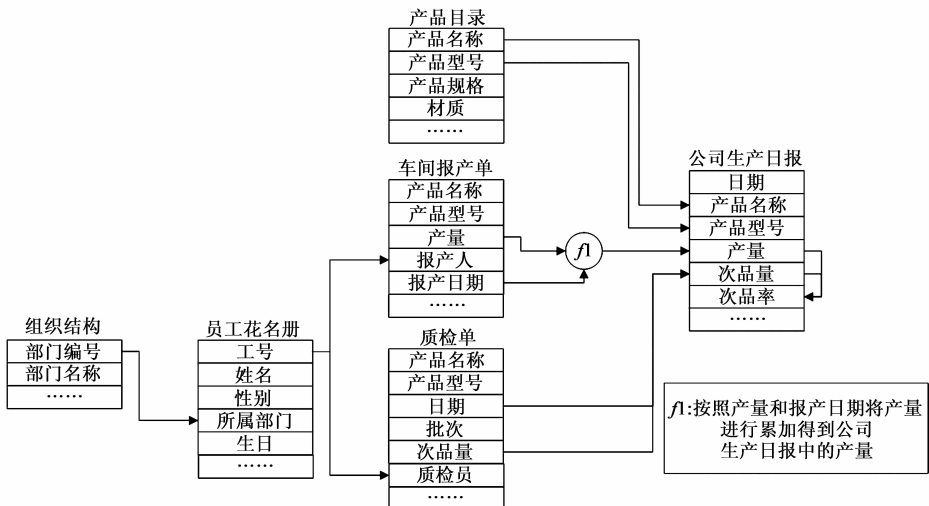


图7 数据源向图 G_1

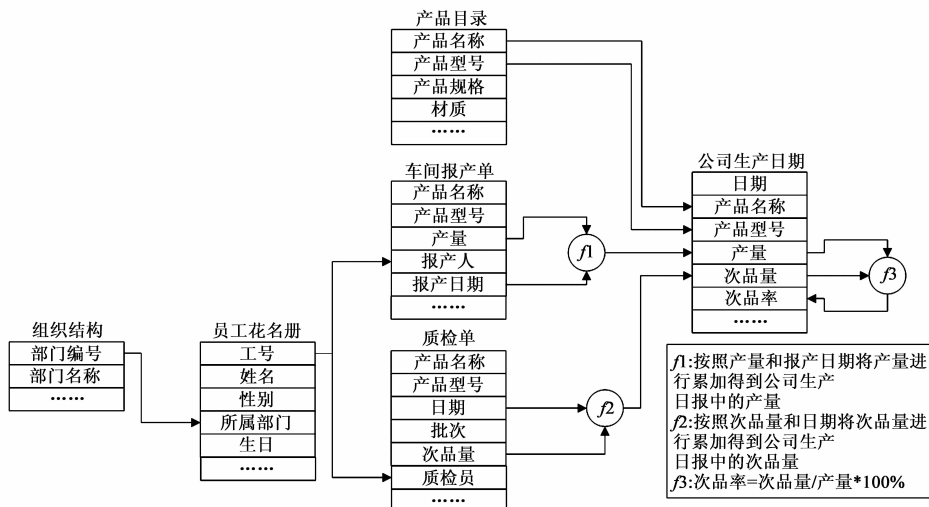


图8 数据源向图 G_3

将是研究数据源向图在业务系统的设计中的应用.

参考文献

- [1] Chen Bingchuan, Li Lei, Wang Heyong. New method of database design based on DFDC[A]. The 2nd International Workshop On Education Technology and Computer Science (ETCS2010)[C]. Wuhan, China, 2010.
- [2] 卢梅, 李树明. 软件需求工程——方法及工具评述[J], 计算机研究与发展, 1999, 36(11): 1289 – 1300.
Lu Mei, Li Shuming. Review of methods and tools of software requirements engineering[J]. Journal Of Computer Research & Development, 1999, 36(11): 1289 – 1300. (in Chinese)
- [3] Lamsweerde V A. Goal-oriented requirements engineering: A guided tour[A]. Proceedings of the 5th IEEE International Symposium on Requirements Engineering (RE' 01) [C]. Toronto, 2001. 249 – 263.
- [4] Patricia Varela, João Araújo, Isabel Brito, Ana Moreira. Aspect-oriented analysis for software product lines requirements engineering[A]. SAC' 11[C]. Taichung, Taiwan, March, 2011. 21 – 25.
- [5] Baniassad E, Clements P, Araújo J, Moreira A, Rashid A, Tekinerdogan B. Discovering early aspects[J]. IEEE Software, 2006, 23(1): 61 – 69.
- [6] Pamela Z, Jackson M. Four dark corners of requirements engineering[J]. ACM Transactions on Software Engineering and Methodology, 1997, 6(1): 1 – 3.
- [7] Sutcliffe A. Scenario Based requirements engineering[A]. Proceedings of the 11th IEEE International Requirements Engineering Conference (RE'03)[C]. California, 2003. 320 – 329.
- [8] 谢海涛, 孟祥武. 适应用户需求进化的个性化信息服务模型[J]. 电子学报, 2011, 39(3): 643 – 648.
Xie Haitao, Meng Xiangwu. A personalized information service model adapting to user requirement evolution[J]. Acta Electronica Sinica, 2011, 39(3): 643 – 648. (in Chinese)
- [9] 陈小红, 尹斌, 金芝. 从交互情景出发的需求捕获方法及其支撑工具[J]. 计算机学报, 2011, 34(2): 330 – 341.
Chen Xiaohong, Yin Bin, Jin Zhi. An approach for capturing software requirements from interactive scenarios [J]. Chinese Journal of Computers, 2011, 34(2): 330 – 341. (in Chinese)
- [10] Svoboda C P. Structured Analysis[M]. IEEE Computer Society Press: 1990. 218 – 237.
- [11] Cameron J R. An overview of JSD[J]. IEEE Transaction Soft Eng, 1986, 12(2): 43 – 46.
- [12] Yourdon E, Coad P. Object-Oriented Analysis[M]. 2nd Ed. New Jersey: Yourdon Press, 1991.
- [13] Booch G et al. Software Engineering with Ada[M]. 3rd Ed. Edwood City, Calif: Benjamin /Cummings, 1994.
- [14] Jacobson I. Object- Oriented Software Engineering: A Use Case Driven Approach[M]. Reading, MA: ACM Press, 1992.
- [15] 利锦标, 李童, 刘林. 基于类图语义框架的中文需求分析方法[J]. 电子学报, 2011, 39(3A): 94 – 98.
Li Jinbiao, Li Tong, Liu Lin. Chinese requirements analysis based on class diagram semantics[J]. Acta Electronica Sinica, 2011, 39(3A): 94 – 98. (in Chinese)
- [16] Kedian Mu, Weiru Liu, Zhi Jin. A blame-based approach to generating proposals for handling inconsistency in software requirements[J]. International Journal of Knowledge and Systems Science, 2012, 3(1): 1 – 17.
- [17] 陈小红, 尹斌, 金芝. 基于问题框架的需求建模: 一种本体制导的方法[J]. 软件学报, 2011, 22(2): 177 – 194.
Chen Xiaohong, Yin Bin, Jin Zhi. Ontology-guided requirements modeling based on problem frames approach[J]. Journal of Software, 2011, 22(2): 177 – 194. (in Chinese)
- [18] ZHENG Yun-xiang, WAN Hai, LI Lei. A new software requirement method based on subject, predicate and object Logic [A]. Software Process Workshop 2005 (SPW2005)[C]. Beijing, 2005.

作者简介



陈冰川 男, 1975 年 12 月出生于河南省新乡市, 现为中山大学信息科学与技术学院软件研究所博士研究生, 主要研究方向为数据库与知识库、软件工程。

E-mail: chbingch@mail2.sysu.edu.cn



吴向军 男, 1965 年生于安徽无为, 中山大学软件研究所博士, 研究方向: 数据库及其应用, 数据结构与算法设计, 软件工程等。

E-mail: issxjwu@mail.sysu.edu.cn