

数据库通讯

SHUJUKU TONGXUN

(数据库设计方法学与设计工具专刊)

2

1984

地 质 矿 产 部

石油地质海洋地质局数据库项目办公室

目 录

数据库设计方法学与设计工具

DATAID项目中数据库设计方法学与设计工具.....	(1)
-----------------------------	-------

第一部分 数据库设计方法学DATAID—1..... (7)

第一章 需求收集与分析.....	(7)
第二章 视图的概念设计.....	(20)
第三章 视图汇总.....	(46)
第四章 CODASYL型与关系型数据库的逻辑设计.....	(67)
第五章 CODASYL型DBMS的物理数据库设计.....	(90)
第六章 物理数据库设计中选择二级索引的一种基于可分离性的方法.....	(113)

第二部分 数据库概念设计工具..... (125)

第七章 NLDA: 用于数据库需求分析的一种自然语言推断系统.....	(125)
第八章 数据库模式与事务处理概念设计的一组统一工具.....	(138)
第九章 INCOD—DTE: 数据、事务处理与事件的交互式概念设计系统.....	(159)
第十章 DIALOGO: 使用语言GALILEO进行概念设计的一个交互环境.....	(183)

数据库设计方法学与设计工具

S. Ceri等著

郭宜斌 译 徐立华 校

(华北计算技术研究所) (地矿部北京计算中心)

DATAID项目数据库设计方法学与设计工具

1. 前 言

本书介绍DATAID项目前三年活动所取得的结果,内容分两大部分,分别反映了该项目的两个主要研究方向的现状。

第一部分描述DATAID—1,这是关于数据库设计的一套方法,这种方法的成功之处就在于它的一般性和它的完整性,因为它适用于好多类数据库管理系统而且该方法学包括了数据库设计的所有阶段。不过,在目前阶段该方法学尚未得到自动设计工具的支持。

第二部分描述由DATAID项目中某研究小组所开发的用于数据库概念设计的某些工具。这些工具也是受到同一设计原则的激发而产生的,这个原则是向数据库设计者(或者,更好的情形是向数据库的用户)提供一种“好的”设计环境。设计工具在这种环境中收集静态和动态定义,检查它们的相互一致性并且有时探测或解决某些设计问题。

2. DATAID—1设计方法学

DATAID—1方法学包括数据库应用在其全部开发过程中的各类设计,该方法学还反映了在[6, 9, 10]中所给出的四个设计阶段:

- 用户需求的收集和分析
- 概念设计(也就是视图设计和汇总)
- 逻辑设计(也就是实现设计)
- 物理设计

该方法学的一般结构见图1所示。

DATAID—1方法学要求对应用环境已经作过某些分析。特别是该方法学并不包括数据库应用的可行性研究和代价—收益分析。该方法学还假定开发数据库应用的企业已经分成了若干个组织环境(视图),每个环境将产生自己的需求说明。

在需求分析和收集阶段将对数据库应用的自然语言描述加以检查、过滤和分级。在这一阶段还需要组织收集专门术语,术语信息又分成数据信息、操作信息和事件信息三类。

用自然语言表达的应用需求

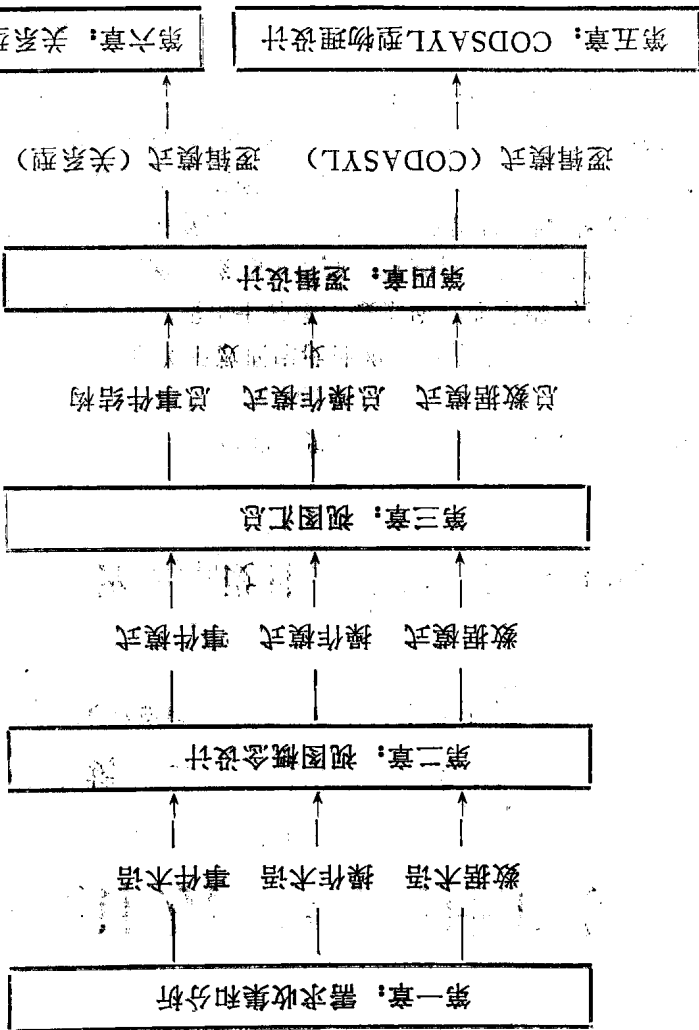


图1. DATAlD方法的总结结构

视图模拟阶段是从以应用的专业术语表达的非结构化信息开始的，经过视图概念设计而产生结构化的模式。对于DATAlD方法学来说至关重要是将数据模式、操作模式和事件模式看作概念模式定义的元素。本方法学是对每个视图独立地全成这个任务的。

视图汇总阶段将数据模式集中到一个唯一的总体模式中，该汇总阶段还产生操作模式的

一个过程性说明，这个说明描述操作的逻辑存取路径，汇总阶段还对所感兴趣的事件产生一个统一的描述，此统一描述即所谓的总体事件模式。

该方法学的这些阶段并不依赖于设计数据库应用时所使用的具体的数据库管理系统或软件支持环境。相反，逻辑设计则一定要针对所使用的数据库管理系统类型的不同而分别进行。

DATAlD-1的当前版本提供了对于CODASYL型和关系型两种数据库管理系统的逻辑设计，物理设计则要反映具体数据库管理系统的物理设计目前适用于

CODASYL定义[4]（这一定义已在若干个实际可用的系统，例如IDMS(Culliname)，

DMS (univac), IDS (Honeywell) 上实现了) 这种物理设计还适用于系统R一类的关系数据库系统[2]、本书有两章专门分别讨论CODASYL环境和关系环境, 其中头一章描述了一种用于确定CODASYL记录和系的存取方法以及它们到存储区域的分配的方法学, 后一章则在定对关系早已方配了主索引后, 如何选择二级索引。值得注意的是这个问题也适用于CODASYL系统。

DATAID—1 独具如下一些特点:

1. 由于它涉及很广泛的数据库系统的所有设计阶段, 因而具有一般性和完整性。
2. 对概念设计提供了统一的语义模型。该模型是实体联系模型[3]的更为丰富的扩充而且容纳了诸如一般化层次结构[8]、聚集、标识符和任选性这样的特性。
3. 将需求并行地说明为数据形式、事务处理形式和事件形式。各种定义逐步地可以求精而且在该方法学的不同的级上对定义的彼此相容性作校验。
4. 逻辑和物理设计阶段是和针对性能的映射同时展开的。本方法学为在从概念模式导出的若干不同的逻辑—物理表示中选择出一种“好”的表示提出了一些准则。

但本方法学目前还有一些限制:

1. 本方法学产生了大量的中间文档, 而要保持这些文档的相容性是既困难又耗费的。但这种情况可以通过提供某些用于管理和编辑文档的自动化工具而得以改善(关于本设计方法学中所用到的各种文档和模块, 我们将在本书的文章中陆续引入并加以描述)。
2. 概念设计的诸特性在DATAID研究小组内得到了较之逻辑和物理设计特性更为广泛的讨论, 因此本方法学不够平衡。

为了描述设计的各个阶段, 我们给出了一个具体实例。为了充分比较DATAID—1与在文献中出现其他设计方法学, 我们使用了在82年5月国际信息联合会的《信息系统设计方法学比较研究》会议上所建议的实例[11]。

3. 自动设计工具NLDA, ISTDA, INCOD-DTE和GALILEO

本书的第二部分描述用于需求收集和数据库概念设计的四个设计工具。

NLDA是在米拉诺技术学院电子工程系的米兰技术人工智能计划中开发的。该工具提供了一个能以半自动方式支持需求分析和视图设计的自然语言接口。NLDA可看成如下三个主要功能的序列: (1) 半自动交互式的用户需求预处理辅助功能。(2) 对处理过的用户需求进行综合分析和语义理解的功能。(3) 对概念视图综合的智能辅助功能。NLDA中的综合分析程序是基于—类特殊的依赖文法, 这一综合分析程序发展成既能用意大利语输入又可用英语输入, 语义分析是基于面向目的的理解方法, 即只从上下文中抽取对特定应用所必须输那些语义信息。因此并不产生对一个语句的完整意义的一般表示。和ISTDA及INCOD-DTE类似, NLDA的综合成分是高度交互地使用其他工具于视图设计和汇总的, 该综合分析程序是完全用LISP实现的, 而语义部分和综合部分的LISP实现目前正在开发之中。预处理部分的设计和实现将随后进行。

ISTDA (Interactive Schema and Transaction Definition Aid) 是在米拉诺技术学院电子系设计的一组最富于表达力的数据库设计辅助工具, 用户借助于一组基本语句来定义其需求并且提供一种面向输出的事务处理定义; 这些需求由ISTDA翻译成两个高级语言: SDL (模式定义语言) 和TDL (事务处理定义语言)。这两个语言都是基于所谓的“调

和二元模型”(C—B模型)，而C—B模型乃是经典的二元模型[1]的扩充：前者包括了属性与类型之间的区别这样的性质或者ISA体系(ISA hierarchies)。事务处理定义语言TDL允许给出事务处理的一种高级的功能定义，TDL有很多性质和通常的询问语言相同，但也有一方面很突出：在事务处理定义语言中可以指定一些依赖于在执行时的实际输入的参数。使用模式定义和事务处理定义测试程序STDT可以对模式定义和事务处理定义两个方面加以测试以发现其不相容性和错误，其基础乃是关于SDL和TDL的解释程序。目前，SDL解释程序及SDL子集的解释程序的PASCAL实现业已完成，但ISTDA的解释程序尚未实现。

INCOD—DTE是在罗马大学工程自动化研究所和米拉诺州立大学控制论研究所设计的。这个工具为数据定义、事务处理定义和事件定义提供了一个集中环境，同DATAID—1一样数据也是用扩充的实体联系模型来描述的，因为本工具在概念级（这一级描述所包括的数据）、制导级（这一级描述存取路径）和执行级（这一级测试数据库的初始实现）上都给出了事务处理定义命令，所以事务处理也是在这些不同的抽象级上来描述的。事件由Petri网来模拟并使用事件说明命令来指定。系统收集用户的定义、检测定义之间的正确性并提供出汇总这些定义的命令。当发现了错误或冲突，系统将向用户提供其他解决方案。INCOD—DTE目前正在由DATAID项目的实现者在Honeywell环境中实现。

GALILEO是在比萨信息科学研究所开发的。GALILEO是用于定义数据库应用的一个高级概念语言，该语言使用了若干典型的概念模拟性质，诸如：面向目标的数据库描述（可以有多重目标描述）、约束的说明性定义。该语言还使用了若干典型的现代程序语言的特性，诸如：数据类型、数据抽象以及模块化，本语言的侧重点在于向数据库设计者提供一种数据库应用开发环境，设计者在这种环境中可以用编辑程序来定义某一应用、编译该应用定义、检测该定义的正确性并存储每次设计会话的结果。GALILEO的一个子集早已在系统DIALOGO中用LISP实现了。DATDID计划的另一工业参与部门目前正在用PASCAL实现GALILEO。

图2从向用户提供交互手段的角度比较了这四种方法，这四种方法提供了范围很广的交互可能性：从自然语言的说明直到高级的、使用程序设计语言的说明。与此相适应的是其用户也从使用自然语言交互的非专家用户(NLDA)直到系统程序员(GALILEO)。

从历史上讲，ISTDA是在DATDIA项目中所开发的用于概念设计具的工第一个建议，ISTDA在过去的设计中曾带有某种程度的不明确性，这主要是由于与用户交互的类型以及留给系统来判断的问题的复杂性所致，留给系统来判断的问题涉及到模拟，重新布局以及视图及事务处理的汇总等。有两个主要的困难可用来刻划ISTDA建议的问题，第一个困难是当时尚未开发出由用户命令语言的规范来支持的一种明晰的和模块化的系统结构，这就使得ISTDA的实现很困难。第二个困难是当时模式定义语言和事务处理定义语言不是一种对用户友好的界面，其原因之一就在于用户的高级规范与这两种语言之间的翻译程序的开发。不过，在该计划开发过程中，当时就已经明显地看到专业数据库设计者想要直接使用设计语言而不是使用高级接口。

INCOD—DTE是在充分考虑到ISTDA设计经验后而开发的。因此INCOD—DTE提供了一组明晰的对用户友好的命令和程序设施，而且总的系统结构是经过精心设计的。压缩了要由系统来判断的问题的范围而且简化了交互的类型。INCOD—DTE还具有与DATDID—1方法学的总的结构相一致的优点（例如它们都使用同一数据模型而且INCOD—DTE还包括了作为刻划性质的事件）。

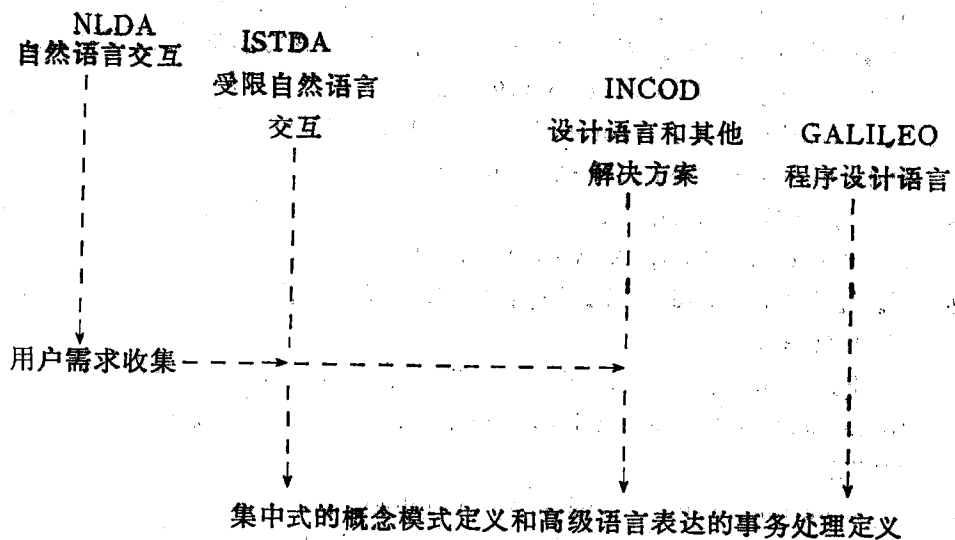


图2. 用户接口的不同的级以及由四个工具所产生的结果

GALILEO是独立开发的。它的设计重点在于力图将现代程序设计语言的性质和“风格”包溶在自己的定义中。作为一种必要的权衡，系统展示了向数据库管理系统性质的一种不大直接的映射。不过，这却引导出一种明晰的设计选择：没有用当前数据库管理系统的不足特别是概念定义手段的贫乏去捆住设计者的手脚。

NLDA是将用完全不同的方法来刻划的，这种方法乃是人工智能的一种典型应用，其目的在于用自然语言与数据库的设计者或用户交互。NLDA对由自然语言表达的描述数据库应用要求的输入稿文执行语言分析和语义理解，语义理解的目的在于在数据库设计的确定的范围内部分地理解这些稿文。视图设计并不是NLDA所主要关心的问题。事实上，视图设计可以通过和INCOD—DTE和GALILEO交互来实现。

在DATDID设计的进一步开发中，重点将是制造INCOD—DTE和GALILEO的工业软件产品、开发逻辑和物理设计阶段的新的工具以及集中现有的所有设计工具。

参 考 文 献

- [1] J.B. Abrial; Data Semantics, in Data Base Management, (W. Klimbie and K.L. Koffeman eds) North Holland (1974) 1-60.
- [2] M. M. Astrahan et al; System R; a Relational Approach to Database Management, ACM-TODS, (1) 2 (1976).
- [3] P. S. Chen; The Entity-Relationship Model; Toward a Unified View of Data, ACM-TODS, (1) 1 (1976).
- [4] Report on the Codasyl Data Description Language Committee, Information Systems, (3) 4, (1978).
- [5] E.F.Codd; A Relational model for Large Shared Data Banks, CACM, 13-6 (1970).

- [6] V. Y. Lum et al: 1978 New Orleans Data Base Design Workshop Report, IBM Res. Rep. RJ2554, and proc. Fifth Int. Conf. on Very Large Data Bases, Rio de Janeiro (1979).
- [7] S. Navathe et al: Information Modelling Tools for Data Base Design, in Data base directions, Fort Lauderdale, Florida (1980).
- [8] J. M. Smith and D. C. P. Smith: Database Abstractions: Aggregation and Generalization, ACM-TODS, (2) 2 (1977).
- [9] T. J. Teorey and J. P. Fry: Design of Database Structures, Prentice-Hall (1982).
- [10] S. B. Yao, S. B. Navathe and J. L. Weldon: An Integrated Approach to Logical Database Design, Proc. N. Y. U. Symp. on Database Design (1978).
- [11] Information Systems Design Methodologies: a Comparative Review, T. W. Olle, H. G. Sol and A. A. Verrijn-Stuart eds., North Holland (1982).

第一部分 数据库设计方法学DATAID—1

第一章 需求收集与分析

1. 前言

本章描述本方法的第一阶段。设计者在本阶段将要对每个组织环境收集其用户需求并将这些需求格式化成数据描述、操作描述、事件描述以及有关的约束。用户需求所涉及的乃是一个组织中计划要由某一数据库应用使其自动化的那一部分。我们将一个组织中在所使用的语言和所执行的功能方面高度相似的那一部分称之为“组织环境”，所谓“组织功能”乃是一组操作以及执行这些操作的条件。为了得到组织环境就必须执行这些操作。

本阶段假定设计者对于组织本身以及我们称之为“契约”的某些内容有很好的了解（不论这种“契约”是写了还是未写）。“契约”规定了数据库设计者的责任，即规定了该组织中预期要使其自动化的那部分的职能。

本设计方法学要求数据库应用规划在数据库设计过程实际开始之前就已经执行，因此，诸如组织情况分析、组织信息流分析、数据库应用的开销/收益分析等等这样很重要的问题并不在本方法学中加以处理。我们还假定必要时可以应用其他的关于组织分析的方法（例如文献1、2、10、11、12所介绍的方法），而DATAID—1方法学所主要涉及的是从关于所要设计的数据库应用的明确的契约开始直到满足契约要求的系统设计成功。

需求收集和分析阶段的结果是：

- 表达数据及其性质的信息需求（数据汇编）；
- 描述对数据必须执行哪些操作的操作需求（操作汇编）；
- 描述对于组织功能起影响的那些事件的事件需求（事件汇编和事件说明形式）。事实上，在本方法学中关于组织功能的形式化描述还必须表示出影响到组织功能的条件以及在组织功能执行之后还成立的条件。正是为了表示条件的变化，我们引进了事件的概念[3]。
- 表达对于数据、操作和事件所设置的限制约束需求（约束需求形式）。

本设计阶段的任务以及本阶段的某些产品在图1中已示明。

本设计阶段的描述按以下三节进行，第2节讨论用户需求的收集。第3节说明如何将用户的需求变换成一种受限的自然语言中的表达式，以及如何将用户需求中的限定条件变成数据/操作/事件/约束语句。最后一节说明如何从限定表达式开始填写数据汇编、操作汇编、事件汇编和事件说明形式。

在本书以后的章节中，我们将使用一个在1980年国际信息联合会关于“信息系统设计方法学比较性研究”的技研委员会所建议过的练习[8]作为一个实例来描述我们的设计方法学。这个实例涉及到一个能支持国际信息联合工作会议组织的数据库的设计。

2. 自然语言的需求收集

第一个设计活动就是确定数据库应用的设计过程的范围。这一活动包括确定与EDP有

关的组织功能并形成一组实际上包括在这些功能中的一组组织环境。

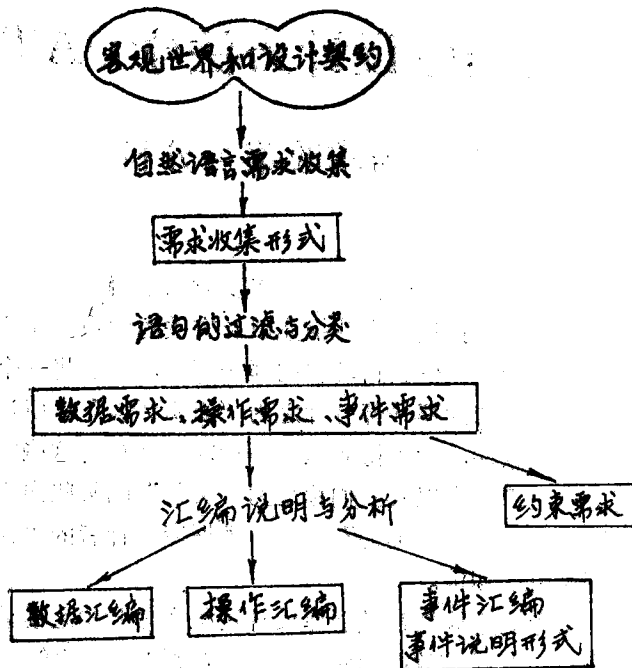


图1.需求收集与分析阶段任务图

关于这一设计计划的总的目标在契约中是这样说明的:

“……所要设计的这个信息系统必需既能支持程序委员会的活动又能支持组织委员会在安排IFIP工作会议时的活动。这两个委员会可看成是一个使用某些共同信息的联合结构中的两个组织实体，这个信息系统应该支持这两个委员会的下述活动……”。

从这个契约以及会议组织的某些经验，我们可以抽取出某些环境和功能并将之填入下面的功能—环境交叉引用表中:

功能—环境交叉引用

页号: 1, 文本: 1, 日期:

功 能 环 境	规 划	收 集	访 问	组 织	邀 请	监 督
程 序	×	×	×			×
组 织				×	×	×

图2. 功能—环境交叉引用形式

2.1 需求收集计划

在这一设计活动中，要将在契约功能中所包括的用户加以标识并列入用户—功能交叉引用形式中（见图3）。

这里的所谓“用户”乃是指该组织中的任何一个人。这个人可以是：

- 实际参加某些功能的人（会议秘书就是参加邀请活动的人）
- 需要有关某一功能信息的人（例如程序主席就需要关于鉴定方面的信息）
- 能够提供（或写出）某一功能在组织中是如何执行的信息或描述。

选择适当从他们那里能够收集到某一数据库应用的需求信息，这对于总的设计计划是一项极为重要的事情。正确选择准则应该是选择这样的一组用户，他们对于一个组织内的全部功能各有很好的了解同时又只具有最小重叠。显然，为了选择出有助于产生出需求规范的一组用户，必须考虑到人的因素和能促进事业发展的因素，这些因素在该组织内部应该是具有典型意义的。

作为一个实例，可以选择如下的用户：

会议主席、程序主席、组织主席、审查评价主席、程序委员会、组织委员会、审查评价委员会、会议秘书处。这些用户在该组织环境中所起的作用由以下的用户—功能交叉引用形式来说明：

用户—功能交叉引用				
环境：组织		页号：1，文本：1，日期：		
用 户 功 能	组织主席	组织委员会	会议秘书处	会议主席
组 织	×	×	×	
邀 请	×	×	×	
监 督				×

图3. 用户—功能交叉引用形式

2.2 需求收集

一个组织内部所有功能的用户需求要通过访问用户、广泛询问、考察用户实际执行功能的况情并从已有的业务文档来收集。本书所介绍的方法学并不区分一个功能是已有的还是计划中的。对于已有的功能，用户要描述目前是如何执行这一功能的；对于计划中的功能，用户应该给出关于该功能应该具有的性质的描述。总之要侧重知道的是系统应该提供给用户什么而不是要了解系统是如何提供的。

在需求收集，对于每一个功能的描述应该指出该功能的目的和该功能所应 包 括 的 操作。特别对每个操作还必须指出；

- 该操作的目的是什么
- 在什么条件下执行该操作
- 如何执行该操作
- 该操作是否总具有应该具有的性质；若不具有这种性质，是在什么条件下使得该操作具有了另外的性质。

设计者还应对操作的执行次序感兴趣，另外，设计者还要得到有关收集对象的生成、修改、校验和删除的信息。

对于每个功能的每个用户，要填写一个需求收集形式，这种形式乃是用一种自然语言书写的关于该功能的视图描述。要记下提供信息的用户以便做可能的回访。图 4 就是针对我们前面的实例而给出某些需求收集形式。

需求收集

环境：组织

功能：监督

用户：会议主席

页号：1

文本：1

日期：

IFIP工作会议是一种国际会议，其目的是将来自IFIP成员国的专家召集在一起讨论对某些IFIP工作小组特别感兴趣的某些技术问题。通常的程序是先有一个并非对每个人都公开的邀请会议，这种会议要保证邀请到IFIP工作小组和技术委员会的成员，不论这些人是否能来，另外，同样重要的是要保证到会的人数足以取得会议收支上的平稳而不超出所允许支配的最大量。

IFIP工作会议的方针要求指派一个组织委员会去处理财政事务、会议安排、邀请和宣传事宜。这个委员会显然需要密切与程序委员会协同工作。这两个委员会需要某些共同的信息并应保持它们所记录的信息的相容性和最新性。

需求收集

环境：组织

功能：组织

用户：组织委员会

页号：1

文本：1

日期：

准备一份要邀请到会的人的名单。优先发布对国家级代表、工作小组成员和联合工作小组成员的邀请。保证每篇为会议所选用的文章的作者都收到邀请。避免对任何个人发出重复的邀请。产生最终的出席人员表。

环境：组织

页号：1

功能：邀请

文本：1

用户：组织委员会

日期：

按邀请名单给每个人发出邀请，避免发出重复邀请。记录邀请的接受情况。

图4. 需求收集形式样本

一旦所有需求都已经收集到了，则应对之加以说明和分析，需求说明就是用一种明确的方法和标准化的方法来对所收集到的需求所作的登录，需求分析应决定需求的完整性和相容性。

[4、5、6、7、9、]

在本书所介绍的方法学中，上述目标是通过语句过滤与分类，汇编说明和分析等任务来达到的。在执行这些任务时，设计者需要有关用数据、操作、事件和约束所表达的用户需求以及这些概念的性质的知识。

3. 语句的过滤与分类

这一任务的目标在于按照适当的规定约束诸功能的自然语言描述(自然语言过滤)，并在被限定的描述内对诸语句的性质作出说明，指出涉及到数据的、操作的、事件的或约束的是哪些语句(语句的分类)。

自然语言过滤

这一活动是按下述方式进行的：

- 验证需求收集形式中说明文字的可理解性以及无歧义性，还可能提出正确性问题以及什么是对用户更为明确的说明；

- 按照适当的规定约束使用的自然语言。关于这种文字说明的安排以及自然语言的约束，我们建议能遵从下述规则：

- a) 将若干个同义的表达式归化为一个表达式；
- b) 通过引入适当的替换而解决同名结构的问题；
- c) 用所指代的名词去代替代词；
- d) 使得不明确元素明确化；
- e) 减少重复和冗余；
- f) 对冠词、数量词、单复数等加以限制。

作为这种过滤活动的一例，试看语句：

“通常的程序（并考虑到当前的目的）是一个并非对每一个人都公开的邀请会议”。

这个语句经过滤就变成：

“一个IFIP工作会议是一个被邀请的会议”。

语句分类

这一活动乃是从被过滤的需求中分别抽取出涉及到数据的、操作的、事件的或约束的语句并按一种特定的形式将之集中起来。正像从图5可以看到的那样，诸语句按其环境而分属于数据的、操作的和事件的三类；事件语句又进一步按其所属的不同功能而分为子类。

需求收集形式中的说明文字似乎是太复杂,这些文字从形式上是可以再细分成一些简单语句的,但无论如何必须保持最初的次序(偶尔情况例外)并保持其逻辑联系。

数据语句乃是借助于数据的性质、该数据与其他数据的联系以及有关基数和具体值的信息来表达关于该数据的所需求的知识。尽管将需求收集形式中的说明文字反映成数据语句主要是归于一中一般的理解过程,但我们还是可以通过诸如:涉及到、是一个、包括在…中、具有、在…内等动词来指出某些“表面标志”。注意,这些动词全是具有“结构”意义的动词。

操作语句表达有关用户如何对数据进行修改或观察、每个操作的频度、操作的执行方式(联机/批方式)等知识、在这些语句的表面标志中,我们可以发现像:准备、发出、寄存、保证、发送、生成等动词。也就是说,这都是一些具有“动态”意义的动词。对于每个操作,形式上都存在一个描述该操作的语句。我们约定:凡表达操作的动词必须是使用祈使语气。

事件语句乃是借助于条件和操作来表达有关一个或几个事件(多个事件也叫事件块)的必须的知识。因此事件语句除了使用与操作语句相同的表面标志外还使用诸如:当…时,一旦…就…、如果、只要等时间和条件表达式。由于有可能对某些条件不存在局部标志,所以这些局部标志必须是或通过考察较大范围的上下文或是通过明确指明的方式重新存储起来

一个事件语句可以伴随多个同时成立的或排他成立的条件以及同时执行的或排他执行的多个操作建立起来。在事件需求形式中,并发性必须通过连词“和”(and)来表达;“排他性必须通过连词“或由”(or instead)来表达;容他性必须通过连词“或也由”(or also)来表达。为了表达一个操作序列,必须写出多个语句并指明其顺序的条件。按照我们的约定,一个事件语句是用一个类似于生产式的格式来表达的,其中条件和操作用冒号分开而操作的动词使用祈使语气:“选择文章;将所选的文章按会议分组并且(AND)选择每个会议的主席”。

约束语句表达在所分析的组织中对数据、操作或事件所施加的限制。约束语句的表面标志是使用诸如必须(不必)、不能等情态动词,即带有“规范”意义的动词。例如:“每篇文章必须至少被三位他人所引用”。

数据需求

环境: 组织

页号: 1, 文本: 1, 日期:

-
- D 1 IFIP工作会议将所有IFIP成员国的专家召集到一起。
 - D 2 IFIP工作会议讨论一个或多个IFIP工作小组所特别感兴趣的某些技术问题。
 - D 3 IFIP工作会议是一个被邀请的会议。
 - D 4 如果某些工作小组(或技术委员会)是在会议组织小组的同一个技术委员会中或者是讨论他们所感兴趣的课题,则这些小组或技术委员会组织在一个会议中。
 - D 5 被邀请的人不出席会议是可以的。
 - D 6 一个IFIP工作会议有一个财政平衡点,因此只能有有限的人参加会议这个最大限度的人数是确定的。
 - D 7 每个IFIP工作会议要指定一个组织委员会。
-

操作需要		
环境：组织	页号： 1， 文本： ， 日期：	
O1	准备被邀请人名单	
O2	优先发布对国家级代表、工作小组成员和联合工作小组成员的邀请	
O3	准备邀请所有其它论文已被录用的作者	
O4	准备邀请所有其论文不被录用的作者	
O5	记录邀请及接受邀请的情况	
O6	准备最终的出席者名单	
..		
事件需要		
环境：组织	页号： 1	
功能：组织	文本： 1	
用户：组织委员会	日期：	
E15	：准备被邀请人名单	
E16	已经准备好被邀请人名单：发出优先邀请	
E17	已经发出了优先邀请：保证其论文已被会议录用的所有作者每人收到一份邀请	
E18	已经保证了其论文被会议录用的所有作者每人收到了一份邀请：保证其论文未被录用的所有作者每人收到一份邀请	
E19	已经保证了论文未被录用的所有作者每人收到了一份邀请：活化对于应被邀请的所有人的邀请	
E20	邀请已经完成：产生最终出席会议人员的名单	
事件需求		
环境：组织	页号： 1	
功能：邀请	文本： 1	
用户：组织委员会	日期：	
E21	：向应被邀请人名单中的一个人发出邀请以避免重复邀请	
E22	所发出的邀请已经得到了接受邀请的回答：记录该邀请已被接受	

图5. 数据操作事件需求形式样本

4. 汇编说明与分析

各种需求的一种明确的和标准化的说明是通过将数据语句、操作语句和事件语句按其源出环境变换成相应的汇编 (glossary) 而得到的。为了填写这些汇编, 有必要对需求收集形式作一扩充还可能要向用户询问更进一步的信息。

当填写汇编时以及在汇编完成之后, 设计者必须验证完整性规则及相容性规则是否被满足。这样的校验可以使得本方法学相当安全, 尽管它受到由不同的用户所给出的不同的需求的影响。

在数据汇编中, 对于每一个数据都要求说明其一个个的具体性质和环境性质。另外, 还必须给出包括该数据的那些数据语句。(见图 6)

所谓数据的具体性质乃是属于该数据所特有的一些特征而与其他数据无关。数据的具体性质包括: 标识代码 (必须具有)、数据名 (必须具有)、一般描述、域 (该数据的取值集合) 和数据的样本值。数据的环境性质乃是相对于其他数据而刻划该数据的。数据的环境性包括: 可能的同义性、该数据通过某个子集联系而与之发生关系的一个或几个数据的名子。

在操作汇编中, 对于每一个操作应指出如下性质: 标识代码、名 (该操作语句的一种有明确意义的缩写, 该名应保留最必要的动词)、操作描述、对操作语句代码的引用、操作类型 (R = 读, I = 插入, D = 删除)、执行类型 (OL = 联机, B = 批方式) 以及参与该操作的数据库所在的数据段。对于每个数据, 作用性质 (Role property) 说明该数据在当前所说的操作中是否具有输入作用 (I)、输出作用 (O) 或被访问作用 (V)。由于某些性质是估计性的, 所以是可变的。

对于事件需求形式的每个事件语句, 事件汇编的一行 (见图 8) 应填入下述性质: 对应于该语句的事件块的标识代码, 和对于该语句代码引用, 对于事件块所引用的功能的引用, 对于在其相应的汇编中所出现的数据和操作的引用 (在操作汇编中引入引用可能会要求该汇编中和事件说明形式中事件说明的修改)。最后, 如果需要的话应该给出不同事件块之间优先关系的指示信息。

对于每一个事件语句还要填入一个事件说明形式 (见图 9) 事件说明形式应该指出下述性质:

- 环境名和事件块所属的功能的功能名。
- 事件块代码。
- 关于该事件块的条件和操作的信息。

对每个条件应当指出下述信息:

- a) 代码
- b) 语义类型, 这种语义类型区分时间条件 (WHEN) 和纯粹的偶然条件 (IF)。
- c) 实际类型 (Pragmatic type), 这种类型按条件的来源而对条件加以分类; 确切地讲就是区分出作为当前事件块所属功能内某些操作的结果而成立的条件 (TRANS)、作为在该功能之外或在从外部插入信息后某些操作的结果而成立的条件 (EXT) 以及对应于日历状态而成立的条件 (CAL)。

对于每一个操作应指出下述信息:

- a) 操作名

b) 操作的实际类型，即将操作按其性质分成观察性操作(OBS)和修改性操作(MOD)。

最后，语法类型按其结构对条件和事件块的操作加以区分：基本条件或操作(EL)和复合条件或操作(与事件语句中的“and”，“or also”和“or instiead”相应的是AND，OR和OREX)。

汇编分析：

为了防止错误的传播和扩大，应该在本方法学的每一阶段进行完整性校验和相容性校验。在需求收集和需求分析阶段，这种校验的目标就是得到包含数据、操作和事件的形式说明的文件。这些文件就是相应的汇编和事件说明形式。这些校验有两种类型：内部汇编(intra)和相互汇编(inter)，这两种情形中，校验都是逐个环境执行的，因为环境之间的校验是属于视图汇总阶段的任务了。

数据汇编

环境：组织

代 码	名 称	描 述	域	实 例	引用	同义	子集
D001	IFIP工作会议	讨论技术问题的国家会议	X(20)	IPIFWG 8.1 意大利	D1	D006	
D002	专 家	具有某方面专门知识的人			D1		
D003	课 题	专家们所从事的技术专题			D2		
D004	IFIP工作小组	召集专家			D2 D4		
D005	国 家	专家们所生活的地方 或举行会议的地方			D1 D2		
D006	被邀请会议	并非对每人都公开的会议			D3	D001	
D007	技术委员会			TG.8 D4 D7			
D008	财政平衡点	会议的总开销		9(6) \$ 50000 D6 O15			
D009	受限制的参加者	所允许参加会议的人 的最大数目		9(3) 200 D6 O15			
D010	被邀请的人	应邀参加会议的人		D5			
D011	组织委员会	处理财政问题		D7			
D012	作 者	宣读论文的人		O3 O4	D010		

图6. 事件汇编一例