

Uniface: 高级第四代应用开发和运行系统

涂方登 徐泽同

涂 健

(中国科学院数学所) (铁道科学院电子所)

摘 要

Uniface是一种高级的第四代语言环境,它使DB应用系统的开发完全独立于具体的数据管理软件。本文还涉及了Uniface的体系结构,介绍了它的主要功能及其将来。

一、引 言

随着数据处理技术的进步,文件、层次、网状、关系等形式的数据管理软件相继问世,加之机型、操作系统的差别,数据处理已面临数据管理软件千差万别的困境,怎样摆脱这种困境已成为当前数据管理的一大问题。对此国内外学者正在进行研究,现已有多种方式的结果,其主要为:

(1) 转换法:主要是研制转换软件,将一种数据管理功能转换为另一种,从而实现应用数据的继承。很显然,这样太专一,难于适应多样化的局面。

(2) 标准化:每当出现一类数据管理软件,人们必然就注意了它的标准化。目前SQL的标准化占有独特的地位,即将推出的我国SQL国标、军标便是这方面的工作。

(3) 统一的应用界面:对新创的数据管理系统而言,遵从标准化当然很好,可现有很多非标准的产品已广为使用而难于丢掉,于是人们便想到了在它们之上再建一层统一的应用界面,如本文介绍的Uniface。

Uniface是依ANSI/X3/SPARC建议,将DB体系结构划分为概念模式、外模式、内模式而设计和实现的一种高级4GL环境。它使DB应用系统的开发完全独立于DB管理软

件和OS,它有良好的用户界面,高度的灵活性和满意的响应速度。Uniface产品大大提高了应用系统开发的生产率,因此值得我们介绍和应用。

二、Uniface的体系结构

ANSI/X3/SPARC的DB研究组把DB划分为概念模式、外模式和内模式结构。概念模式是DB的整体逻辑描述,即DBA见到的DB;外模式是用户看到和使用的DB;内模式是物理上的DB。荷兰Uniface B.V.依上述思想于1989年设计实现了Uniface,它的体系结构如图1所示。

“系统管理员”用IDF(信息工程与设计)定义应用系统的概念模式,“应用设计员”在上述概念模式基础上使用IDF定义用户的外模式。这些定义均被集中存于“数据字典”中,此数据字典本身仍为支持Uniface的DBMS管理的一个DB。“运行系统”用数据字典中的定义来控制末端事务,在底层的内模式进行工作。“外部结构编辑”是Uniface的末端用户界面,是一种功能齐备的全屏幕编辑,末端用户通过它与外模式定义的应用级DB对话,进行有关的DB操作。“DBMS驱动器”以DBMS本身的语言与概念模式中每个实体指出的DBMS对话,内模式将实体映射

为一个或多个DBMS的表或文件,从而实现了Uniface对DB的独立性。“概念模式管理”主要管理数据字典中的概念信息,并协调运行系统中各成分的工作。

本质上应用设计负责创建特定的数据管理任务;运行系统负责管理临时工作空间中由外模式处理的数据;而DBMS则管理磁盘

Uniface用框架(frame)表示数据元素,每种元素都有对应的框架。设计员可决定场框架,展现场的任何部分内容。框架移动和编辑功能允许用户浏览及编辑框架上的数据。实体框架并不展示所有事物,而压缩了数据表示。设计员可决定同时在框上出现多个事物。Form用上述各种框架来表示数据结构,它和结构编辑是Uniface交互式应用的重要成分。

3.2 IDF功能

IDF以菜单方式供系统管理员和应用设计员使用,它有下列选项:

概念模式(Conceptual Schema):供系统管理员定义概念模式。它通过访问Form来定义实体、场、域、实体的关系、全局定义模型等。

外模式(External Schema):供应用设计员定义外模式。

外模式被编译为运行时使用的高效表格,允许测试外模式打印定义的文档、查阅外模式使用的所有处理、从应用字典中删除不用的定义等。

字典列表(Dictionary Listings):打印应用字典中信息的交叉引用列表,这些列表给出了IDF指明的诸项定义的概观。

应用(Application):定义应用级。比如定义第三代语言、DBMS驱动器、信息区位置等。也用于连接应用中引用的例行程序。

中心定义(Central Definitions):用于定义处理库、集中管理的公用处理程序。还用于定义信息的中心稿文系统、多种语言的呼救稿文,并确定替代键盘的转换表等。

SQL工作台(SQL Workbench):向开发者提供从支持SQL的DBMS终端直接发出SQL命令的能力。

杂项(Miscellaneous):含有稿文文件编辑、语言配置、代码列表、许可权管理

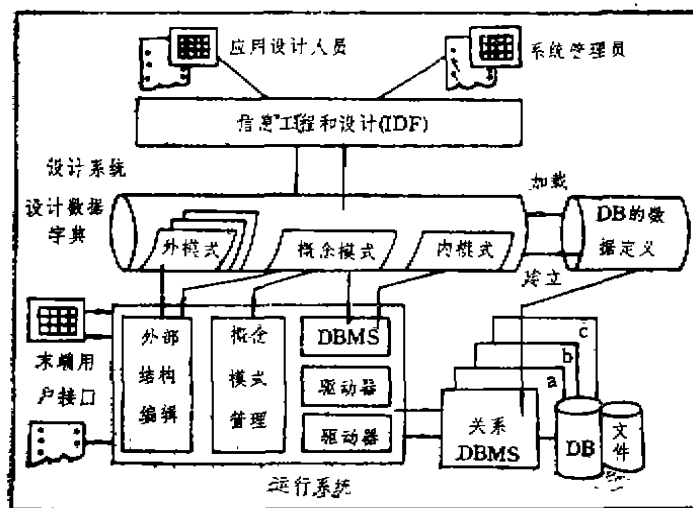


图1 Uniface体系结构

a.多键ISAM; b.层次DBMS; c.其它。

上由内模式指定的数据,并在运行系统指导下工作。^[1]

三、Uniface各成分的功能

前面已介绍了Uniface的体系结构,现较细地介绍其各成分的功能。

3.1 Uniface的数据结构及表示方式

在Uniface中,使用了下述数据结构:场(field),它对应于关系DB中的属性或列,场含有Uniface中有效的信息;

实体(entity),它对应于关系DB中的关系或表,或对应于其它类系统中的文件,它表示DB中可识别的对象;

事物(occurrence),是实体中诸场的实在值的集合,它对应于关系DB中的元组或行;

子型实体(Subtype entity),访问另一实体的别名,它有其自己的定义和关系,但引用它对应的超型实体的场和键。

等。

3.3 定义概念模式

系统管理员用IDF定义概念模式，对DB的场、实体、实体间的关系(Relationship)进行集中描述，因此这种DB描述是完整而抽象的。下面进一步介绍描述过程。

(1) 定义实体 实体是DB中可识别的对象。系统管理员常在概念模式定义实体的Form上定义实体。一次可将多个实体赋给一个概念模式。当定义实体时指定该实体的DBMS十分重要，因为DBMS决定了该实体由这DBMS管理。Uniface支持一个概念模式中的各实体可由不同的DBMS管理，并可动态改变DBMS。除DBMS而外，Uniface完全与DB无关。

(2) 定义场 一个实体有若干个场。系统管理员若将光标移到定义实体的Form上的Field Assgn处，并按DETAIL键，则可进入定义该实体的场的Form。其后系统管理员则可在该定义场的Form顶部定义关键字，并在其底部定义各个场。关键字由一组场名组成，它唯一标识了该实体。Uniface保证对主关键字(Nr1)不会被修改。各个场由场名、域、Db性质、类型、接口、语法、格式等项内容组成。

(3) 定义实体间的关系 在概念模式定义里，实体间关系的定义是重要的，这常以Form形式来做。系统管理员可将一实体的主关键字移入另一实体，作为两个实体相关联的外部关键字，从而可定义实体间的“一”对“多”关系。每个关系还可定义可引用的完整性约束，确定当“一”实体被删除时“多”实体将如何办？

若Uniface不能而DBMS却能检查某事，则它可生成DDL去定义一约束。Uniface不管各实体由不同的DBMS存贮，均能自动实施这些约束。

另外子型实体也能用于定义不同角度使用的关系，比如它允许定义实体内“一”对“多”的关系、“多”对“多”的关系、实体间的

“多”重关系等。

系统管理员还可使用IDF装入功能，从DBMS和CASE工具库中装入已有的定义来定义概念模式。

3.4 定义外模式

Uniface并不生成专门的软件去处理特定的任务，在Uniface应用中所有处理均由广义的数据定义指明的约束来完全定义。

在概念模式中系统管理员已定义了概念模式的实体和场，也指明了它们的若干值，但留下某些定义的互斥控制，其它的为满足个别事务需能被设计员增加和修改。于是概念模式定义的缺省值成了设计员定义外模式的起点。

(1) 数据驱动和被触发事件的处理 在Uniface应用中，约束处理由每个实体和场的定义来决定。数据驱动这些处理。处理是概念和外部数据结构中固有的，多数约束在概念模式中定义，余者则由外模式增强和修改。

场和实体定义决定了事务中处理的数据的性质和约束等，这些性质的定义或以声明(declaration)的形式指明，或作为处理说明(specification) proc来指明。proc由事件触发，当与proc关联的触发器被触发时，则proc活化。这意味着处理由数据定义来定义，被用户界面来的事件活化。Uniface提供了40余种触发器，它允许设计员精确指明事务会话时何种事应发生。这些触发器在场、实体、Form和应用级有效。

(2) 外模式定义的工具 外模式定义包括可见的Form，各种处理和定义事务的说明。因此一个外模式定义包括画Form、Form级定义和场及实体的定义等。Form级定义包括定义Form窗口的大小、位置、外模式的类型、边界等。在该Form的上部支持声明的定义，下部定义处理说明(proc)的三类触发器、即通用触发器、会话触发器、I/Q触发器。

画面(paint tableau)是设计员“画”外

部结构的场面。Uniface的作者为创作性的屏幕设计提供了广泛的可能性,并保证了在屏幕上见到的即是逻辑数据模型的如实视觉表示。设计员不仅用画面画了Form的外形,而且也确定了它的外在结构。Uniface使您看到了什么即定义了什么。

框架用FRAME功能设置。设计员将光标移到希望之处并按FRAME,则在那儿安设了实体或场的框架。设计员在A实体的框架内安设B实体的框架时,他不仅决定了B的位置,而且也告知了Uniface A、B有上述关系。框架在画面上的相对位置是整个Uniface需要理解的外部结构。

快速Form允许设计员将概念模式中与一实体关联的若干场或全部场快速地加载到画面上,并用这些场作为画外模式的基础。在定义布局时快速Form也十分有用。

(3) 场和实体定义 设计员常用画面去指明以概念模式提供的定义为基础的场和实体的定义。当设计员把光标移到画面上场框架处并按DETAIL时,则出现场定义的Form。实体定义方式也类似。

若干定义集中于单一模型(model)名之下,常用的各种标准检查则可按此名引用,于是可定义场的语法模型,场的布局模型等检查功能。总之约束定义由参数,定义模型和proc组成。每个模型和proc控制了场或实体处理的一个特定方面。

在场和实体级允许定义多种触发器,它们决定了在指定的场或实体里一特定的事件出现将发生何事?有四种触发器:它们是语义验证和计算、场提示序列、会话控制、I/Q等方面的触发器。与之相关则定义了相应的proc。

(4) proc语言 proc语言是4GL语言,可决定何事件对何数据发生而应作何事。proc包括高级语句,它用单个语句可启动复杂的处理。公用的proc可置于中央proc库,该库中的proc能被任何触发器按名调用。proc语言有下述功能:

条件: if, else, endif;

赋值和计算: (assignment), compute;

信息: message, askmess, putmess, clrress, help;

事务控制: commit, rollback, store, retrieve, erase;

I/O控制: read, write, delete, lock, reload, open, sql, print, release, lookup;

分支和连接: goto, label, perform, spawn, run, call, entry;

出口: return, done, exit, opexit, end。

事物控制: creocc, setocc, remocc;

结构编辑: edit, display, \$ prompt;

ASCII文件操作: file-dump, file-load,

调试: debug, nodebug;

杂项: clear, blockdata, refresh,

unmset, numgen, set, reset,

\$ keyboard, \$ prompt, \$ variation。

借助sql指令,SQL可视为proc语言的扩充成分。同时proc语言还可引用C、COBOL、FORTRAN等3GL语言。

3.5 数据库接口

Uniface通过DBMS驱动器直接与多种DBMS接口,从而充分利用了多种DBMS的功能。现已开发的驱动器允许Uniface与关系、层次、DBMS、索引顺序访问方式(ISAM)的系统以及顺序文件系统相接口。设计者还可用“Uniface DBMS Driver Cookbook”按其所需开发新的DBMS驱动器。当前Uniface已有ORACLE、Rdb、Britton-Lee、INGRES、Focus、dBase III/IV、RMS、SYBASE、SQL Server、C-ISAM、TRIP/TDBS、SQL/RT、INFORMIX、BASIS及顺序文件驱动器。正在作Share Base驱动器。

(1) uniface的驱动器 驱动器是Uniface运行系统和特定DBMS之间的媒介。IDF生

成设计员定义的应用，而DBMS 驱动器则是产生可执行程序的部分。

每当需要I/O,运行系统就调用适当DB-MS的驱动器，该驱动器翻译Uniface的通用功能为DBMS 完成所要求操作的指令序列。而每当要实施DBMS的I/O时，Uniface 发出调用DBMS I/O 的功能。该调用以填入可用数据结构的要求码来寻找驱动器的前台地址，而驱动器则检查要求码，并转向处理特定要求的子程序，于是Uniface便启动了驱动器工作。

在驱动器接口中有多种DB 操作（如op-

en, close, updata, select, fetch, commit等）的Uniface功能；有启动及终止DB-MS 对话的功能；有识别 DBMS 特征的功能等。

(2) 检索、编辑和贮存数据 下面用例子来解释Uniface怎样同时与两种不同的DB 进行接口，并检索、编辑和贮存数据。在此例中用关系DB 管理 Contpers和Visits, 用ISAM保存Corresp文档，用Uniface Form描述外部结构，用框架内的框架描述和定义数据结构。该例子有下述外模式和内模式（图2）；

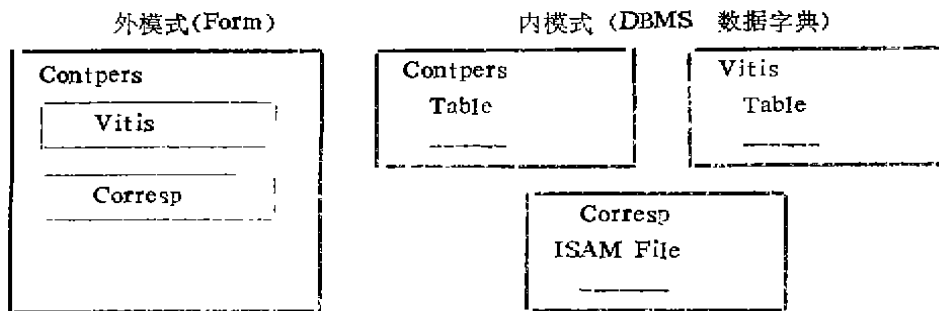


图2 外模式和相应的内模式

实体的关系			概念模式RBASE				
One entity	to entity	Many entity	Index y[N]	Foreign order of	Key fields in prim./cand_Key	Key Number	Constraints Del Upd
CONTPEERS		CORRESP			RELSNAME, PERSNAME	1	CAS RES
CONTPEERS		VISITS			RELSNAME, PERSNAME	1	CAS RES

图3 通过外键定义的实体关系

实体间的关系在概念模式中定义。Visits 与Contpers由Pelsname和Persname场联系，因Uniface 包含数据结构的概念模式。当编辑时为保持DB 的完整性（甚至越过不同的DBMS），Uniface 将检查引用的完整性约束。当用户删一公司的记录时，设计员能决定是否要删除该公司的联系人。下面是概念模式中通过外部键而定义的实体间的关系（图3）。

为了检索特定联系人的数据，用户打入一名字（主关键场）并按RETRIEVE 即可。用户也可用“Form 查询”。为了检索名中有

“J”开头的联系人，则在NAME场打“J*”并按RETRJEVE。之后一个驱动器发出必要的SQL命令，从一关系DB的两个表（关系）中检索联系人和访问的数据，另一驱动器发出ISAM I/Q 语句去检索SIAM 文件中的联系人的信件。从DB 中检出的东西保留在工作存储器中，用户可用结构编辑编排这些外部结构中的数据。

在一典型事务中用户检索数据，然后增，删，再按STORE则对DB 发送修改，当然也可按QUIT 而放弃修改。存贮修改分两步进行：a, Uniface分析外部结构，判定有无增、

删、修改事物；b.创建一内部事件为修改或插入事物触发Write proc；为删除事物触发Delete proc。这些proc包含调用适当驱动器子程序修改DB的语句，这些驱动器产生并执行所要求的修改。

(3) Uniface的事务处理 Uniface中事务处理发生在应用启动和第一个“Commit”之间以及两个“Commit”之间，它允许设计员精确定义怎样去构造事务。事务的精确定义允许Uniface用十分直截了当的并发控制机制来保证不同用户不能同时修改同一数据。当用户要修改事物时，多数DBMS提供了封锁事物的机制，而Uniface则用了利用这些功能的proc指令。驱动器将上述proc指令编译为特定DBMS的DML命令，于是DBMS的封锁功能支持了Uniface的封锁功能。

Uniface有两种封锁，即谨慎锁和乐观锁。当用户首个修改某事物的数据时，则锁定此事物。若该事物已被另一用户锁上，Uniface则发信息到用户，用户可继续编辑，然后再回来处理此事物。若一事物已由另一用户删除，则向用户发信息；若事物未被删，也未被封锁，Uniface则将DB中的事物与外部结构上的事物比较，以确保Form上的数据仍是现在的。若不匹配，即另一用户修改了它，则Uniface发信息到用户，且读一新事物到外部结构并锁上；若数据仍是现在的，则Uniface封锁此事物并进行处理，锁定直到执行完“Commit”或“rollback”指令为止。这类锁通常用在需要最大保护且事务频率低

的环境。

乐观锁推迟封锁事物直到最后可能时刻，这在多用户环境中有很大好处而且在两用户同时修改同一事物的低概率环境中是理想的。乐观锁不顾“lock”指令，而是在“写”或“删”指令活化时，Uniface自动封锁一事物。在要判定一事物是否仍是现在的时，即它是否被另一用户修改或删除了时，Uniface要进行检查。若事物不是现在的，则用户会收到一出错信息，在必要时用户应重新启动此事务。若I/O操作时出了问题，则Uniface会自动通知DBMS将事务退回到上次“commit”指令执行之处。

四、Uniface的将来

Uniface至今仅接受了集中式DBMS作为支持，因而走向网际，支持分布式数据处理是它将来的发展。图4展示了它实现分布式数据处理的网络策略。因Uniface用C写成，易于移植。在此策略中将运行系统一分为二，主要部分置于Uniface工作站上（即顾客处），小部分（Uniface服务器和驱动器）则置于大型机上。当然设计系统也在Uniface工作站上。

Uniface应用处理访问一事物时，工作站通过网络将结构描述信息送向大型机的服务器，每个I/Q操作引用这些信息。服务器指挥DBMS驱动器从各DBMS管理的DB中获取数据。然后送回发生请求的工作站。每个工作站能使用一台机器上的服务器，也能使用多台机器上的服务器。

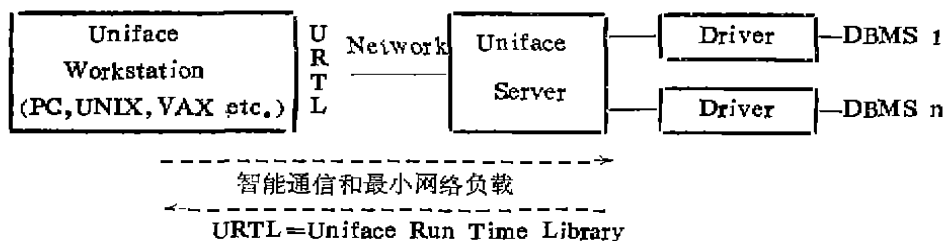


图4 Uniface的网络支持

除上面外，还有与已具有分布数据处理能力的DBMS和DDBMS相接口而加上分布数据处理能力的方法，如现在正在做的Sh-

are Base驱动器和我们设想的Ada-DDBMS驱动器。Share Base是以数据库计算机作为服务器、多种类通用数字计算机作为顾客、

OODB存储管理子系统的设计^{*)}

李伟华

胡守仁 (国防科技大学计算机系)

摘 要

GOODbase是我们正在 SUN 工作站上开发的一个通用的面向对象数据库系统, 本文简单介绍了该系统的总体结构及其存储管理子系统的逻辑结构, 详细描述了它的内存管理的实现方案, 包括对象存储器的划分、对象数据结构的建立、对象的引用、内外存对象的交换等方面的设计。GOODbase存储管理系统的设计充分考虑了对内外存对象的快速访问。

面向对象数据库 (OODB) 是程序设计语言、数据库技术、软件工程、人工智能以及非传统数据库应用等方面研究的共同产物, 它从八十年代中期一出现就显示了强大的活力, 并迅速形成了世界范围的研究热潮。总的来看, OODB是一个程序设计语言与数据库的集成系统, 它能很好地支持上述各个方面的应用, 但主要的特征仍然体现在数据库功能方面。绝大多数OODB都可以不太严格地划分为两个主要子系统, 即存储管理子系统以及一个运行于其上的解释器, 存储管理子系统是诸如永久性、并发控制、恢复、一致性和查询等数据库功能的最基本的实现支持。

如同传统数据库一样, 在OODB中, 所有的数据对象都存放在辅助存储器上, 处理时必须送入计

算机主存储器中。为此要在主存中设立若干管理表格, 开辟若干个系统缓冲区, 作为数据库系统的所有系统模块与辅助存储器之间的虚拟存储器接口, 在这个接口上, 可以直接访问数据对象。存储管理系统就是负责这些管理表格的维护、缓冲区的管理、外存对象的存放、对象在主存与外存之间的移动、新对象的创建、权限的设置、索引的建立、恢复、并发控制等等, 并提供高层模块所要访问的对象。

由于面向对象数据模型的思想与传统的关系、层次、网络模型的思想有实质性的区别, 所以在面向对象数据库系统中, 要引用不同的对象表示与存储结构, 以及更为复杂的引用机制, 从而对存储管理系统提出了很高的要求。

中经通讯网连接而成的硬软结合的分布式使用的RDBMS体系结构。Ada-DDBMS是我们正在研制的 VAX、VMS、DECNET 运程网上的Ada 相关的分布式关系型数据库管理系统。这种作法对 Uniface 而言是增加了分布式的数据管理能力; 对Share Base及 Ada-DDBMS而言, 则是增强了用户界面的功能。为Uniface 做一个适应于 ShareBase 的驱动器, 或做一个适应于Ada-DDBMS 的驱动器将是不难办到的, 我们正在为此努力创造条

件。最后衷心感谢陈兴明, 邢晶先生为我们提供了资料!

参考文献

1. Introduction to Uniface, Uniface B.V. Amsterdam 89.4
2. 徐泽同, 涂健, 共享库概述, 计算机科学 90.6期
3. 徐泽同, Ada相关的分布式数据库管理系统 Ada-DDBMS, 计算机科学 90.1期

^{*)} 国家自然科学基金与863高技术资助研究项目