第36卷 第19期 计 算 机 工 程 2010年10月
Vol.36 No.19 Computer Engineering October 2010
·软件技术与数据库· 文章编号：1000—3428(2010)19—0107—03 文献标识码：A 中图分类号：TP311.13

通用虚拟维修开发平台中数据库的应用

贾 π[1]，陈新来[1]，徐晓刚[2]，吴 晶[1]
(1. 海军蚌埠士官学校信息技术系，安徽 蚌埠 233012；2. 海军大连舰艇学院装备系统与自动化系，辽宁 大连 116018)

摘 要：数据库技术在通用虚拟维修开发平台设计中具有重要作用。基于此，分别从数据库表的设计、程序与数据库之间的通信和Web数据库服务器的设计3个方面对通用虚拟维修开发平台数据库进行设计。提出数据与逻辑分离的数据库设计思想，利用Apache、PHP和MySQL作为数据管理开发工具，通过TGE三维开发引擎实现对数据库的访问，从而实现对系统数据的控制管理。实验表明，将数据库技术应用在虚拟维修开发平台中，能提高开发平台的通用性。

关键词：通用；虚拟维修；开发平台；数据库

Application of Database in General Exploitation Platform for Virtual Maintenance

JIA Pai[1], CHEN Xin-lai[1], XU Xiao-gang[2], WU Jing[1]
(1. Department of Information Technology, Naval Petty Officer Academy, Bengbu 233012, China;
2. Department of Equipment System & Automation, Dalian Naval Academy, Dalian 116018, China)

【Abstract】The function of database technique is very important in designing general exploitation platform for Virtual Maintenance Training System(VMTS). The database is designed from the designing of the database tables, the communication between program and database, the designing of Web database server. The database designing idea that database separates from logic is offered. Apache, PHP and MySQL are used as database exploitation tools, and the communication between program and database is realized by TGE 3D engine, finally the controlling of the database is realized. Experimental result indicates that using the function of database technique in general exploitation platform for VMTS makes it more general.

【Key words】general; virtual maintenance; exploitation platform; database

1 概述

针对目前大部分虚拟维修训练系统存在通用性差、可扩展性不足的问题，文献[1-4]在系统通用性设计方面做了大量工作，内容涉及通用性结构设计、人机交互、视景仿真等关键技术，其设计思想对通用虚拟维修开发平台设计具有一定的参考价值，但实现开发平台通用性的难点是如何实现具体装备信息与系统程序之间的分离。

虚拟维修过程中存在着大量的具体装备信息，如模型属性、维修规则、故障状态和维修任务信息等，其数据类型也比较复杂，包括模型、图片、动画、文本等。这些信息放在程序中工作量大，日后不便于进行管理。而建立数据库后通过人机交互界面即可以方便地对数据进行添加、删除和修改等操作，为数据管理提供方便的技术支持，同时缩短了开发周期，实现了系统的易维护性和通用性。

本文针对开发平台的通用性难题，采用数据库开发技术，提出了“数据”与“逻辑”分离的设计思想，将装备和维修过程信息存储于数据库中，当重新生成另一装备的虚拟维修训练系统时，只需在装备建模后修改数据库中的相关参数即可实现另一型装备的虚拟维修训练，无须再大规模修改程序，实现了开发平台的通用性。

2 通用虚拟维修开发平台数据库的设计

2.1 数据库表的设计

为了实现系统的通用性，根据系统数据需求，设计了以下数据表：

(1)任务信息表：存储不同的维修任务编号、任务名称、维修设备名称、任务描述、维修序列编号。(2)任务零件表：存储不同任务所对应的具体零件编号、零件名称、空间坐标、初始状态及约束关系。(3)零件表：存储零件类型、数量、所对应的维修工具以及零件模型存储路径。(4)维修规则表：描述零部件之间的约束关系，通过设定其约束关系实现对零件拆装顺序的描述。(5)工具表：存储工具名称、编号。(6)故障现象显示表：存储单个现象的类型、模型存储路径、现象约束关系。(7)故障树表：包括故障树节点对应的零件编号、节点的子节点组成、零件状态、节点层级，用于存储整个故障树信息。(8)维修日志表：存储每次训练的时间、用户姓名和得分等信息。(9)用户信息表：保存可以使用本系统的所有用户的相关信息。

2.2 程序与数据库之间的通信

程序与数据库之间的通信主要是对数据的访问。在系统程序运行之前需要先向数据库发出连接请求，若成功建立连接则可进行数据查询。

基金项目：辽宁省自然科学基金资助项目(20082176)；浙江大学CAD&CG国家重点实验室开放课题基金资助项目(A0906)
作者简介：贾 π(1982－)，男，助教，主研方向：虚拟现实；陈新来，副教授；徐晓刚，教授；吴 晶，讲师
收稿日期：2010-04-24 **E-mail：**pi_jia@163.com

2.3 Web 数据库服务器的设计

架设 Web 数据库服务器，将数据库置于广域网下，操作人员通过虚拟维修训练系统客户端可实现异地访问，能解决资源重用的问题；将数据存放在服务器中，能保证数据的安全；数据库管理人员通过网页完成对数据的录入编辑操作，维护方便。

Web 数据库服务器由三部分组成：数据库，数据编辑网页和网络服务器。通过数据库管理软件实现对数据的管理；利用网页编程工具开发数据编辑网页；使用 Web 服务器软件作为网页访问数据库的桥梁。

根据开发平台需求，设计了 4 种不同功能的数据编辑网页：装备数据编辑，维修任务编辑，日志查询和用户注册网页。4 种网页中数据编辑操作所提供的数据可满足系统数据设计需求。

3 通用虚拟维修开发平台数据库的实现

3.1 开发工具

为了实现开发平台的通用性，根据数据库设计要求，选择 Apache、PHP 和 MySQL 作为数据管理开发工具，通过 TGE[5]三维开发引擎访问数据库，从而实现对系统数据的控制管理。MySQL 作为数据库管理系统，负责数据的存储、访问以及数据修改；利用 PHP 开发数据库编辑交互界面；使用 Apache 构建网络服务器，作为网页访问数据库的桥梁。此开发模式具有开发成本低、速度快、功能强大等优点。

3.2 数据库的建立

首先在数据库访问页面创建一个新的数据库，再根据数据库表设计要求逐个添加设计好的数据库表。在创建表前需要确定表的结构，并进行优化，为数据模型选择合理的存储结构和存取方法，决定存取路径和分配存取空间等。数据库表建立以后，即可在表中输入数据。

3.3 程序与数据库之间通信的实现

数据库存储了零部件、故障和维修规则等信息，在虚拟维修操作过程中要从数据库实时提取以上信息，需要程序与数据库之间进行通信，实现的过程如下：

(1)建立程序与数据库之间的连接。程序启动时发出建立与数据库连接请求，打开 MySQL 数据库，初始化 MySQL 对象，将数据库地址、用户名、密码以及数据库名称等赋给该对象；若连接成功则可进行数据查询。

(2)对数据的查询与操作。系统根据需要调用不同的查询语句对相关信息进行查询，并将查询结果存储在内存中。如查询所有的维修工具信息：首先查找维修工具表，再将表中所有数据暂存在一个数组中，逐次将每个工具的工具名称、编号、模型存储路径等信息取出，从而得到所有工具信息。

(3)断开连接。当程序退出时，程序发出断开与数据库连接请求，断开后程序与数据库之间的通信结束。

4 通用虚拟维修开发平台数据库关键技术

通用虚拟维修开发平台开发的关键是实现“数据”与“逻辑”的分离，将具体装备信息存放在数据库中，如零件数据、故障数据、维修规则等；把不因具体装备变化而改变的控制响应固化在程序中，如场景显示、故障现象显示、维修操作响应和维修任务生成等。

4.1 零件数据与场景显示

加载零件数据并在场景中显示的实现过程如图 1 所示。用 SQL 语句在零件表和任务零件表中查询零件，找到所有符合条件的记录，提取相关信息；使用 datablock 语句定义零件数据块；通过 new 语句实例化零件对象；最后用任务对象的句柄调用 Add 函数将零件显示到场景中。从模型加载流程可知，只要将零件数据存储在数据表中，不用修改代码即可实现零件在场景中的显示。

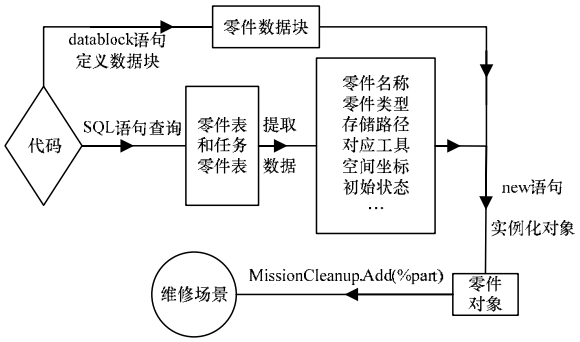


图 1 零件加载流程

4.2 故障数据与故障现象显示

故障模拟主要通过可变纹理、粒子系统、播放动画、3D 音效等来实现。如何实现故障现象显示和故障数据的关联是实现故障模拟的关键。

(1)存储故障文件名及其相关属性。将故障文件名及其相关属性存储在表 1 所示故障现象显示表中。故障的显示与隐藏由其约束零件及其状态控制，系统对约束零件及其状态进行检查，符合故障现象显示表中的约束零件状态条件时故障就会显示或隐藏；对于受多个约束零件状态控制的故障现象，提取所有约束零件的状态，最后对故障现象显示做出判断。

表 1 故障现象显示表

字段	属性
ContentID	零件编号
FalutTypes	故障现象类型
FalutName	故障文件名称
Display	显示/隐藏
ID	约束零件
Status	约束零件状态

(2)故障现象显示。故障文件名存储到故障现象显示表中后，只需在脚本中指定故障文件所在的目录，通过基于索引的查询即可找到目录中的故障文件，再将故障模型载入维修场景中，通过操作响应实现故障现象的更新。

加载故障模型时，首先确定故障现象依附的零件编号(ContentID)，根据故障现象类型(FaultTypes)确定添加动画、改变纹理、粒子系统或音效模型。实例化零件对象后，将数据库中该零件的每个字段的数据即属性值都传给该对象，并将每个实例化的对象保存到一个数组里，可以通过对象的句柄来调用其属性值。以某个零件模型为例，该零件出现故障时，在数据库中用故障现象类型即 FaultTypes 字段值来判断添加何种效果；用变量%FaultType 来存储从数据库中读进的 FaultTypes 字段的值；实例化模型对象，对象句柄赋值给变量%item；将%FaultType 值赋给%item. FaultType；用简单的 switch、case 语句判断%item. FaultType 的值，不同的值做不同的动作，如若值为 0 就播放动画，值为 2 就为模型改变纹理，3 为添加粒子，4 为添加音效，从而实现故障的模拟。

4.3 维修规则与操作响应

由于维修动作和维修对象具有响应的因果关系，因此使用因果关系作为一种规则即可较简便地实现维修操作正确性的判断。“if…then…”的判断形式可实现对于维修操作的描述，即“如果做了什么操作，即会有什么结果”。由于不同装

备具有不同的维修规则，因此为了满足开发平台通用性的需要，需将具体的维修规则放在数据库中。

(1)维修规则的存储。只要建立零件之间的相互约束关系就能够解决设定维修规则的问题，在数据库中建立维修规则表以存储维修规则。维修规则如表 2 所示。

表 2　维修规则

字段	属性
ContentID	所选零件编号
AccessoryState	零件初始状态
rState	零件结果状态名称
ConditionID	约束零件编号
ConditionState	约束零件状态
MissionID	任务编号

(2)维修操作合法性判断。维修操作合法性判断标准由维修规则决定，维修规则作为数据存储在数据库中，程序从数据库中读取维修规则，最后与实际维修操作进行比对，对操作合法性进行判断。维修操作合法性判断流程如图 2 所示。

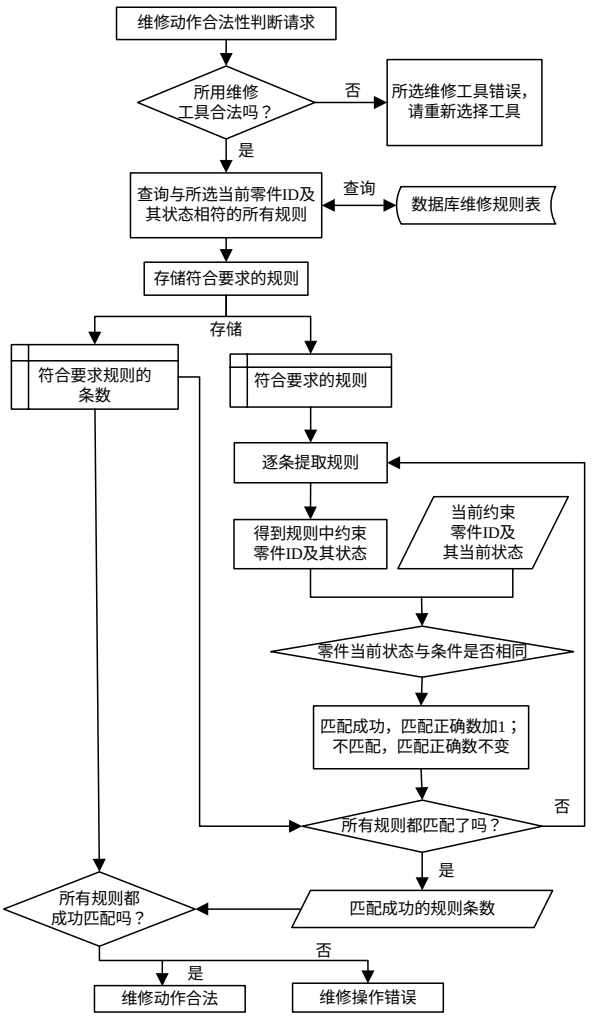


图 2　维修操作合法性判断流程

4.4　故障树与维修任务生成

故障树是一个故障现象诊断过程的逻辑结构，它的根节点对应着故障现象，故障树的每个分支对应着导致该故障现象的一个可能的原因，最终故障原因表现为故障树中的一个叶子节点。利用故障树的这一特性，可以通过建造故障树根据零件故障状态来定义维修任务。

(1)故障树表的建立。故障树如表 3 所示，零件状态指零件处于装上还是卸下的状态；节点层级标识节点是否为根节点；零件类型可分为“好”、“坏”和“备用”3 种状态。

(2)故障树的存储。根据装备技术说明书上提供的维修步骤，结合零部件状态，存储故障树。

(3)维修任务生成。故障树从根节点到叶子节点有很多条路径，每条路径相当于一个维修任务。这些任务都是由相应的零部件状态组成的，这些状态即为装备在维修场景中的初始状态，最后的叶子节点为装备修复后的状态。将每条路径上节点编号记录下来，存储在维修任务表中，即得到了多个维修任务。根据用户选择的不同维修任务，加载不同的任务数据。

表 3　故障树

字段	属性
ID	编号
ContentID	零件编号
ChildID	子节点编号
Status	零件状态
Desc	零件当前状态描述
Types	节点层级
AccessoryType	零件类型

5　实例应用

根据通用虚拟维修开发平台的标准要求，利用三维开发引擎和数据库开发工具建立了一个桌面式通用型虚拟维修开发平台。利用该开发平台，在 3 个月时间内成功开发了某型舰炮和某型雷达自动标绘仪 2 种不同装备虚拟维修训练系统。系统实现了实时交互，在 P4 1.8 GHz/内存 1 GB/256 MB 显存的笔记本上，速度不低于 34 f/s。图 3 为某型雷达自动标绘仪维修操作界面。



图 3　利用维修工具对零件进行拆卸的场景

6　结束语

本文利用“数据”与“逻辑”分离的数据库设计思想，初步实现了开发平台的通用性，并在缩短开发周期的情况下减少了劳动强度，提高了开发效率。但随着现代装备的发展，所面对的维修操作越来越复杂，在数据库表的设计、人机交互编辑数据的方便性以及实现的关键技术等方面仍需进一步研究。

参考文献

[1] 王晓光, 苏群星. 虚拟维修通用仿真软件系统的设计[J]. 计算机仿真, 2006, 8(8): 266-268.

[2] 方传磊, 苏群星, 刘鹏远, 等. 导弹装备虚拟维修训练系统通用平台[J]. 计算机工程, 2009, 35(3): 274-276.

[3] 冯宇晨, 刘金林, 曾凡明. 通用虚拟维修训练仿真软件的设计与实现[J]. 中国修船, 2008, 8(4): 42-44.

[4] 刘金林, 曾凡明. 基于 CATIA/VIRTOOLS 的舰船主动力装置虚拟维修训练技术研究[J]. 舰船科学技术, 2008, 12(6): 140-142.

[5] Finney K C. 3D Game Programming All in One[M]. 徐明亮, 谷培培, 译. 北京: 清华大学出版社, 2005.

编辑　任吉慧