

文章编号: 1000-5862(2017)01-0046-06

# 在建模语言 Apla 中实现多媒体数据库应用的方法研究

朱小征<sup>1 2</sup> 薛锦云<sup>2\*</sup> 夏 鲸<sup>1 2</sup> 熊小舟<sup>1 2</sup>

(1. 江西师范大学计算机信息与工程学院 江西 南昌 330022; 2. 江西师范大学国家网络化支撑软件国际科技合作基地 江西 南昌 330022)

摘要: 随着信息技术在各大领域的广泛应用,产生的多媒体数据所占比例越来越大,对多媒体数据处理的需求也越来越多,高级程序设计语言支持多媒体数据处理的功能也变得尤为重要.采用模型驱动的思想,在支持模型驱动软件开发 PAR 平台的 Apla  $\rightarrow$ Java 程序自动生成系统中,扩充对多媒体数据处理的功能,并提出多媒体数据处理机制和 Apla 中基于关系代数的文本处理机制自然融合,包含了对多媒体数据处理的 Apla 语言程序,可以通过 Apla  $\rightarrow$ Java 程序自动生成系统自动转换成等价的 Java 程序,使得 Apla 语言具有对多媒体数据处理的能力.

关键词: PAR; 模型驱动; 多媒体数据处理

中图分类号: TP 311 文献标志码: A DOI: 10.16357/j.cnki.issn1000-5862.2017.01.09

## 0 引言

随着信息技术的飞速发展,计算机应用产生的数据量也越来越大,数据类型十分丰富,如文本、图形、图像、动画和声音等不同类型的的多媒体数据.多媒体技术不是各种信息媒体的简单复合,它是一种把文本、图形、图像、动画和声音等形式的信息融合在一起,并通过计算机进行综合处理和控制的,能支持完成一系列交互式操作的信息技术.正是多媒体技术的发展,使得计算机能够成为一种普通工具并广泛应用于工商、教育、行政、娱乐与家庭生活等领域.多媒体技术在发展,而大多数高级程序设计语言在对多媒体数据的处理能力上仍较为欠缺,高级程序设计语言若不能较好地处理多媒体数据,将会落后于时代的发展而逐渐被淘汰.所以支持对多媒体数据的处理,对于高级程序设计语言来说是极为重要.

目前国内外众多学者基于形式化与自动化研究,设计了相关建模语言,但遗憾的是大多数建模语言不能引入像多媒体数据库处理机制特征.而 PAR 方法和 PAR 平台<sup>[1-44]</sup>(简称 PAR)成功引入了多媒体数据库处理机制,在 Apla  $\rightarrow$ Java 程序自动生成系统中,依托 Oracle 数据库,重新设计并扩展了相关核心模块,使 Apla 语言具有多媒体数据处理能力,

并能通过转换器生成正确的可执行的 Java 程序.

## 1 Apla $\rightarrow$ Java 程序自动生成系统简介

PAR 方法是一种统一的证明和程序设计方法<sup>[15]</sup>. PAR 方法主要有以下几点创新思想和技术:扩充的量词变换规则、统一的算法程序设计方法、基于类型定义的构件组合技术、循环不变式的新概念、新的开发技术、灵活方便的泛型程序设计机制;带关系代数的 Radl,包括 Radl 的相关规约描述说明和程序设计语言 Apla,以及一系列程序转换系统.

PAR 平台主要包含了下列典型的转换系统: Radl  $\rightarrow$ Java 算法程序自动生成系统; Apla  $\rightarrow$ C++ 算法程序自动生成系统; Apla  $\rightarrow$ Java 算法程序自动生成系统; Apla  $\rightarrow$ 其他语言的算法程序自动生成系统:如 C#和 VB 等高级语言.

Radl 语言是一种基于递推关系的算法设计和规约描述语言,由算法规约和描述 2 部分构成,它的作用是描述算法程序相关的规约和变换规则. Radl 语言是为实现 PAR 方法而设计的辅助规约语言<sup>[16]</sup>相当于 MDA 体系中的 CIM 计算无关模型,是顶层抽象语言.

Apla (Abstract Programming Language) 是 PAR 平台下的抽象程序设计语言,同 Radl 语言一样是为

收稿日期: 2016-07-12

基金项目: 国家自然科学基金重大国际合作研究项目(61020106009) 国家自然科学基金面上项目(61272075 61472167) 和国家自然科学基金地区科学基金(61462041) 资助项目.

通信作者: 薛锦云(1947-) 江苏海门人 教授 博士生导师 主要从事软件形式化和自动化的研究. E-mail: jinyun@vip.sina.com

形式化开发算法和程序而设计的. Apla 抽象程序设计语言是 Radl  $\rightarrow$  Apla 算法程序自动转换器的目标语言(输出语言), 又是 Apla  $\rightarrow$  Java, C++, C# 等转换器的源语言(输入语言). 其特征明显符合 MDA 体系的 PIM 平台无关模型, 不依托相关平台. 而 Apla 转换得到的相关目标代码语言如 Java 语言就是 PSM 平台相关模型, 为从平台无关模型 Apla 代码处得到的模型的转化和输入数据.

Radl 和 Apla 语言有较多相似的地方, 如标识符、关键字、过程、函数、数据结构类型以及符号表达方式完全一致. 但 Apla 语言因为偏重程序设计, 故其比 Radl 语言多了程序设计相关方面的内容(过程和函数).

## 2 Java 语言及 Oracle 数据库对多媒体数据的处理

随着信息技术的发展, 日常生活中产生各种数据的类型变得多样化, 多媒体数据类型的数据所占比例也越来越大, 而早期的关系数据库只能处理像字符串等简单类型的数据, 这早已不能满足人们对于多媒体数据的存储需求. 为了解决这个问题, 一些主流的关系数据库已经扩充了对多媒体数据处理的功能, 即在关系模型的基础上增加大对象 LOB (Large Objects) 类型的数据类型. 本文以 Oracle 11g 数据库为基础来介绍关系数据库处理多媒体数据的方法, 以及 Java 语言处理多媒体数据的方法.

### 2.1 Oracle 数据库对于多媒体数据的处理

在关系数据库中, 多媒体数据文件等大对象通常是通过 LOB 类型字段来进行存取的. Oracle V8.0 以及更高版本提供了大对象(LOB)的存储管理支持, LOB 可以分为 4 种: 二进制大对象(BLOB), 字符大对象(CLOB), 民族语言化字符大对象(NCLOB)和文件大对象(BFILE), 其中前 3 种数据类型是将多媒体数据大对象存储在数据库内部, 而最后一种数据类型是将多媒体数据大对象存储在数据库外部. 最常用的是 BLOB 和 CLOB 这 2 种数据类型, 本文实验也是基于这 2 种数据类型而完成的. BLOB(Binary Large Object), 即二进制大对象, 是一个可以存储二进制文件的容器, 存储的二进制文件最大可以达到 4 GB, 图片文件就是典型的 BLOB; CLOB(Character Large Object), 即字符大对象, 是一个可以存储字符文件的容器, 存储的字符文件最大也可以达到 4 GB, xml 文件就是典型的 CLOB. 通过

这 2 种数据类型使得 Oracle 数据库能够存储图形、图像、音频、视频、文档等特殊类型的多媒体数据.

操作 Oracle 数据库所提供的 LOB 的方式也较多, 即可以使用 Oracle 数据库自带的 PL/SQL 工具来操作 LOB, 也可以使用 Java 等高级程序设计语言来操作 Oracle 数据库中的 LOB. 将一个多媒体数据大对象存储到 Oracle 中的 LOB 字段要比将一个普通数据类型的数据存储到 Oracle 中复杂得多. 要将一个多媒体数据大对象存储到 Oracle 中的 LOB 字段中, 首先要创建一个包含有 LOB 字段的数据表, 然后使用 EMPTY\_LOB(·) 函数将一个空的 LOB 对象插入到该数据表的 LOB 的字段中, 接着以更新的方式获得将要插入的 LOB 字段并锁定该行, 最后把要存储的二进制大对象的内容读入到 LOB 字段中.

### 2.2 Java 语言对于多媒体数据的处理

Oracle 数据库在关系数据库的基础上扩充了对多媒体数据处理的能力, 这为 Java 等高级程序设计语言能够处理多媒体数据提供了基础, 如今的 Java 程序设计语言已经扩充了对多媒体数据处理的功能. JDBC 提供的 java.sql.Blob 和 java.sql.Clob 这 2 个接口分别对应于 Oracle 数据库中的 BLOB 和 CLOB 这 2 种数据类型, 可以使用 ResultSet 和 PreparedStatement 接口中的 getBlob(·)、getClob(·)、setBlob(·)、setClob(·) 等方法来操作 Oracle 中的 BLOB 和 CLOB 字段.

在 Java 中, 把一个多媒体数据大对象存储到 Oracle 数据库中的过程, 与使用 PL/SQL 将多媒体数据大对象存储到 Oracle 数据库中的过程是相同的. 以下通过把一个二进制大对象存储到 Oracle 数据库中的 BLOB 字段为例: (i) 使用 DDL 语句创建一个包含有 BLOB 字段的数据表: media: create table media(id int primary key, blobFile Blob); (ii) 使用 EMPTY\_BLOB(·) 函数将一个空的 BLOB 对象插入到该数据表的 BLOB 字段中, (iii) 以更新的方式获得将要插入 BLOB 字段并锁定该行: stmt.execute("insert into media(id, blobFile) values(' + id + ', ' + "'empty\_blob(·)')"); rs = stmt.executeQuery("select blobFile from media where id = " + id + "for update"); BLOB blob = (BLOB) rs.getBlob("blobFile"); (iv) 把要存储的二进制大对象的二进制内容写入到该 BLOB 字段中.

从 Oracle 数据中取出多媒体数据大对象的过程则相对较简单一些, 直接使用 getBlob(·) 方法便能完成操作, 但取出来的是一个 Blob 对象, 还不能直接使用, 需要将其二进制内容写入到相应格式的

文件中才可以使用: `rs = stmt.executeQuery("select blobFile from media"); Blob blob = rs.getBlob("blob-File");`

### 2.3 在 Apla 语言中扩充多媒体数据库操作的前提

Java 语言对多媒体数据的处理并不是完全靠自己来完成的,而是新增加一些接口,然后通过这些接口来操作多媒体数据库的 LOB 字段,而多媒体数据库的 LOB 类型的数据类型是在关系数据库的基础上扩充的,这就使得对多媒体数据的操作能够与关系数据库相融合,正是由于这一点,才使得在 Apla 语言中扩充对多媒体数据的操作功能成为可能。

## 3 多媒体技术在 PAR 平台中的实现方法

该文基于 PAR 扩展和重构部分原先的 Java 生成系统以及融入多媒体数据库技术,实现 Apla  $\rightarrow$  Java 程序的自动转换系统的扩展。PAR 平台自动转换系统运用了模型驱动(MDA)的思想,通过基于算法和程序模型,按照对应模型转换规则,自动生成所需目标模型。

### 3.1 Apla 语言数据库相关操作介绍

在传统程序设计语言中,程序设计语言和对数据库的操作是没有关联的,各类编程语言都是通过 SQL 语句来实现对数据库的访问和处理,而不必自己设计操作数据库语句。SQL 语言是一种标准的数据库操作语言,常需要编写复杂嵌套的 SQL 语句才能完成特定的任务。将这类查询代码嵌入到 Java、C++、C# 等高级程序设计语言当中,不仅不能实现形式化推导和证明,而且会使用户难以理解,也会大幅度降低程序的查询性能。众所周知,SQL 语句是基于关系数据模型而发展建立的一系列标准化的查询语句,而关系代数的抽象程度更高,且有着严谨的数学基础和理论支撑。前文已经提到 Apla 语言的特性就是基于这一点着手,设计和开发了直接通过关系代数的编程方式来执行数据库相关程序,通过对数据库和操作的抽象提取,使得数据库查询语句变成了简单明了的关系代数式,且带有编程语言的功能。不仅提高了程序效率,也提高了代码的可靠性。下面举一个典型的 Apla 语句数据库方面的操作来说明体现对比: Apla 语句: `rs =  $\Pi$ {sname, cname, grade} (  $\sigma$ {sex = '女'  $\wedge$  grade > 90} (B  $\cup$  E))`; 对应的 SQL 语句为: `[select surname, cname, grade from( select * from( select temp1. sid, surname, ssex, sage, cname, grade`

`from B temp1, E temp2 where temp1. s_id = temp2. sid) temp7 where ssex = '女' and grade > 90) temp8 ]`。对比中明显可以看出 Apla 语言对于数据库操作方面的优越性,在形式化推导和代码简洁清晰上优势非常明显。

### 3.2 Apla 多媒体技术

在 PAR 方法的指导下,通过对多媒体技术的研究, Apla 的多媒体技术处理方式基本与已经成功融入的事务处理和异常处理机制处理方式保持一致,即同样的 Apla 语言编程风格,这也体现了 Apla 以及 PAR 平台结构的清晰和可扩展性。

Apla 多媒体技术主要测试和使用了 Oracle 11g 数据库以及相应的 jar 包,用于文件上传和下载的 commons-fileupload-1.3.1.jar, commons-io-2.4.jar。采用 JDBC 的连接方式,摒弃了原先的 ODBC 连接方式。Apla 多媒体技术主要操作文档、图像、音频、视频等相关对象。以视频为例,实现的功能主要为:上传一个或多个视频以供用户保存相关重要数据(文件上传之后保存到相应的后台 Oracle 数据库当中);显示用户所需的一个或者多个视频信息以供用户查阅(从数据库内提取相应的视频目录到前台页面进行显示);播放视频功能(从数据库中提取出视频文件,并通过相关多媒体播放器进行播放);下载一个或多个视频文件以供用户提取相关重要数据(从后台 Oracle 数据库中取出视频文件,并并保存到本地磁盘当中);以及最后的删除相应的视频文件。

### 3.3 Apla $\rightarrow$ Java 程序转换器扩展

PAR 平台下的 Apla  $\rightarrow$  Java 程序生成系统是以 Java 为目标代码的转换器,其完整的运行,不仅仅需要 Apla 语言 and 核心转换器(词法、语法、语义、转换模块),而且还需要功能强大的自定义构件库 strulib 支撑算法程序运行以及 Table 类库支撑数据库相关操作。此次是对原有 Apla 语言机制的扩展实现,基于原先已经设计实现的转换器原型,采用原先的设计思想和方法,对相关模块进行扩展和增添功能。

在原有的 Apla  $\rightarrow$  Java 转换器上进行扩展和增添相关模块工作,使之具有多媒体数据库应用程序处理能力,并修改相关形式化描述,使之能够统一使用一套 Apla 语言机制(例如 connect 语句,从 4 个参数变为系统需求的 3 个参数),而不是重新设计新的 Apla 语句,这样就避免重新设计词法语法等相关模块。因此项目的研发同样也包括原有的词法分析检查 Lexana( Lex. h 和 Lex. cpp)、语法分析检查 Synana

(Syna.h 和 Syna.cpp)、核心转换库 trans(trans.h 和 trans.cpp)、出错处理 errorp(Errorp.h 和 Errorp.cpp) 以及支撑库 table(封装和定义多种操作) 和 strulib 文件库(封装和定义类库)。界面部分 UI 也保持与原先的 Apla  $\rightarrow$  C++ 程序的界面风格一致, 对系统也进行了一些必要的原先不完善之处的补充修改和扩展(包括输入、输出、数据库操作、类型匹配、输入符号的处理优化等), 使系统更加稳定和完善。涉及技术安全问题, 此处只论述 table 库和 strulib 库的扩展工作, 对于核心转换器工作本文不再进行详细论述, 主要工作为添加多媒体数据库相关的关键字和相应的配套操作。如在 Apla 中加入多媒体类型 image、video、audio、source、doc 作为关键字及相应的转换器更新, 同时加入 Clob、Blob、Nlob、Bfile 数据类型。主要使用了 MDA 模型驱动的思想 and 以下技术:

- (i) 使用了文件上传技术, 在客户端将要保存的 Oracle 数据库中的多媒体文件上传到服务器端;
- (ii) 使用文件下载技术;
- (iii) 使用了 JavaBean 技术, 并封装到 strulib 库内作支撑;
- (iv) 使用了 webservice 技术进行传输;
- (v) 以及 Oracle 多媒体相关技术。

**3.3.1 Table.java 类的扩展** 原有 Table 类已经实现对大部分 Java 代码形式的数据库操作语句的支持, 并证明执行方法的操作是正确的, 但是基于转换器的模块增加和功能扩展, 其需要不断的扩充完善进行同步匹配工作。此次扩展主要为 connect、update、write、writeln、select 以及交、插、并、除等一系列基本数据库相关操作, 使其具有多媒体数据库处理能力且兼容原先的基本数据处理能力。此外还增加了 exception、setnull 等方法。主要思想是保持原有结构, 能不增添新的方法和尽量地使用封装技术。

**3.3.2 Strulib 库扩展** 根据 Apla 语言特性以及多媒体数据库应用技术在 Java 中的处理方式, 在类库内增加了 Video 类, 使之能够处理 Java 多媒体数据库技术相关中的接口 Serializable。Strulib 类库中原有较多支撑类(如 arraybag、arraydigraph、btree、edge、edgenode、linkdbag、MathOp 等)。Video 类是一个保存视频相关内容的类, 主要有视频编号(int id)、视频的真实名字(String videoName)、视频保存的名字(String saveName)、视频文件内容(Blob videoFile)、描述(String description) 等几个成员对象以及针对这 5 个成员对象产生的 10 个公有成员函数。

**3.3.3 典型 Apla 代码** 以下代码主要在系统中扩展了原先的相关语句, 使之能够满足多媒体的操作,

使得词法和语法没有变化。这样也符合 MDA 的开发模式, 不需要词法语法变化, 只改变具体的 PSM。以上的代码(如 write、 $\Pi$  等操作)均采用原先的 Apla 语法格式, 但是其意义已经更新扩展, 可以处理多媒体相关的关键字, 而不是原先的只能处理和识别普通类型(如 char、string 等)。

```
Program testmedia;
type good = record //type 结构
id: string;
bl: blob; //新数据类型 blob, 多媒体相关数据
end;
var D: table;
X: good;
rs: ResultSet;
begin
connect( "DB" ,"username" ,"password" );
//统一为 3 个参数
X. id: = "4";
X. bl: = EMPTY_BLOB( );
createTable( D ,"id char( 10) not null ,bl blob ,
cl clob" ); //创建一个表 D;

D: = D  $\cup$  { X }; //添加一组数据;
rs: =  $\Pi$ { bl } (  $\sigma$ { id = 3 } ( D ) );
//选择相关结果存入 rs 结果集中
write( F ,"D: \\test\\3.jpg" );
//输出相关图片操作
X. id: = "4"; //赋值操作
D: = D - { X. id }; //从 D 内删除 X. id 数据
close( "DB" ); //关闭数据库操作
end.
```

## 4 应用实例

选取一个简单的案例进行说明, 操作方法几乎与原先的方法一致, 所有的不同操作均在 table 支撑库内更新, 故只需要把多媒体数据类型当作普通数据库相关数据类型处理即可。除了 insert 时需要增加一个语句, 先插入一条空白语句后再插入相关数据, 而不是直接插入一条数据。案例说明: 从数据库选出 id 为 3 的图片和删除 id 为 4 的多媒体数据。代码如图 1 所示。



- [10] Zheng Yujun ,Xue Jinyun. Object-oriented specification composition and refinement via category theoretic computations [J]. Lecture Notes in Computer Sciences ,2006 , 3959: 601-610.
- [11] Zheng Yujun ,Ling Haifeng ,Xue Jinyun. Ecogeography-based optimization: enhancing biogeography-based optimization with ecogeographic barriers and differentiations [J]. Computers & Operations Research ,2014 ,50: 115-127.
- [12] Zheng Yujun ,Ling Haifeng ,Xue Jinyun ,et al. Population classification in fire evacuation: a multiobjective particle swarm optimization approach [J]. IEEE Transactions on Evolutionary Computation 2014 ,18( 1) : 70-81.
- [13] 王昌晶 ,薛锦云 ,左正康. SRLtoRadl 生成系统及其范畴论语义 [J]. 电子学报 2014 ,42( 1) : 137-143.
- [14] 谢武平 ,薛锦云. Radl 算法到 Apla 程序的生成系统 [J]. 计算机研究与发展 2014 ,51( 4) : 856-864.
- [15] 谢武平. Radl →Apla 程序生成系统及其可靠性研究 [D]. 南昌: 江西师范大学 2009.
- [16] 朱小征 ,薛锦云 ,徐华珍. Transaction 在 PAR 平台中的实现方法及应用研究 [J]. 计算机与数字工程 ,2015 ( 10) : 1884-1890.
- [17] 岳巍. PAR 平台中数据库系统多媒体数据处理的方法研究 [D]. 南昌: 江西师范大学 2015
- [18] 马传松 ,邓静 ,李振坤. 在 ORACLE 数据中如何对多媒体数据对象进行处理 [J]. 广东自动化与信息工程 , 2003( 3) : 40-42.

## The Research on Implementation Method of Multimedia Database Applications in the Modeling Language Apla

ZHU Xiaozheng<sup>1 2</sup> ,XUE Jinyun<sup>2\*</sup> ,XIA Jing<sup>1 2</sup> ,XIONG Xiaozhou<sup>1 2</sup>

( 1. College of Computer Information Engineering ,Jiangxi Normal University ,Nanchang Jiangxi 330022 ,China;

2. The State Base of Networked Supporting Software of International S/T Cooperation ,Jiangxi Normal University ,Nanchang Jiangxi 330022 ,China)

**Abstract:** With the extensive application of information technology in various fields ,the proportion of multimedia data is increasing ,and the demand for multimedia data processing is increasing. Advanced programming languages support the function of multimedia data processing has become particularly important. The purpose of the paper is to extend the function of multimedia data processing in Apla →Java program automatic generation system by using the model driven architecture( MDA) ,which is successfully in PAR platform with the international leading level. The multimedia data processing mechanism proposed and the text processing mechanism based on relational algebra in Apla can be integrated naturally ,and the Apla program which contain the multimedia data processing can be automatically converted into the equivalent Java program by the Apla →Java program automatically converted system ,it makes the Apla language has the ability to process multimedia data.

**Key words:** PAR; model-driven; multimedia data process

( 责任编辑: 冉小晓)