

基于元数据对象动态运行记录系统的设计和实现^①

蔡延华¹ 任 钢² (1.广东电网公司 湛江供电局 广东 湛江 524005;

2.中软公司 华南基础构件中心 广东 深圳 518057)

摘 要: 针对当前电力行业运行日志记录系统软件开发中所面临的需求及环境多变的情况,提出了一种基于元数据对象的架构模型。通过对元数据对象模型的描述并给出其关系模型和各部分组件结构框架,阐述了在该模式下自适应业务的实现方法,使模型组件应对不同领域的应用都具有可配置型、可适应性和可复用性,从而提高系统开发的效率和质量。

关键词: 运行日志记录;元数据对象;表现层组件;业务引擎组件

Design and Application of Dynamic Operation Log Recording System Based on Meta-Data Object

CAI Yan-Hua¹, REN Gang²

(1. Guangdong Power Grid Co., Zhanjiang Power Bureau, Zhanjiang 524005, China;

2. The South-China Component-Based Center of ChinaSoft Co., Shenzhen 518057, China)

Abstract: This paper puts forward a Meta-Data Object Architecture Model for the Variable situation of demand and circumstances in Operation Log Recording software exploitation of Power Industry. It describes Meta-Data Object and gives the Relationship Model and Components Model Structure with various parts. It expounds the self-adaptive business methods based on the model, making the model deal with applications in different fields with configurable, adaptability and reusability, and hence improves the efficiency and quality of the system development.

Keywords: operation log recording; meta-data object; presentation component; business engine component

1 引言

在电力信息化建设中,常常有一些基于值班运行记录的日志管理系统。其运用的场合主要是值班员把在当值工作中的各种情况按照事件类别记录下来,并且是按照事件发生的先后顺序来进行排序。当交接班转移给下一个值班员,下一个值班员做同样的工作。这样形成了一整套运行记录的流水帐日志,保证了整个运行日志管理的不间断性、完整性和延续性。这些运行日志记录也可以作为将来进行电力的统计、分析和管理的的基础数据。在传统的开发模式中,开发人员会根据用户运行记录的要求设计相应的表结构,即一种记录类型对应一个表,然后针对每个表开发出相应的增加、修改和删除模块,然后增加一些查询、统计

和报表等功能。但实际情况是用户的运行记录是多种多样,并且用户的需求也是变化的,针对用户的所有个性化需求去设计不同的记录模块,然后定制组件去满足用户需求的方式不够现实,尤其是企业的发展和市场的变化也会使信息系统的底层数据结构产生变化,采用这种传统硬编码实现的运行记录系统不具备足够的灵活性、移植性和扩张性。任何细微的变化都要求重新调整表的结构和具体业务的重新编写。

本文就是要解决用户需求多样性和变化性等因素而采用元数据对象模型来建立运行日志管理模型,通过元数据对象模型自身的调整能够适应各种业务应用环境的变化,并随之自动做出相应的功能调整,满足客户不断变化的需求。因此能够提高开发效率,降低

^① 收稿时间:2009-05-25

维护难度, 满足变化的需求。

2 系统分析和说明

2.1 系统分析

运行日志管理中, 有很多相关的业务对象, 如运行日志类别、值班人员、日志本(簿)、日志记录、运行日志单位和班组、记录时间、记录人等。但是里面核心的是日志记录。它上面连接着日志簿和类别, 下面有生成日志的具体内容。所有的运行管理系统, 基本上都是以日志记录为基础来进行扩充和完善的。所以, 日志和日志记录是运行日志管理的核心, 是主数据对象。如果能对这些主数据采用元数据来描述, 然后在具体操作过程中也基于元数据来实现, 那么就能自适应用户需求的变化。

元数据^[1,2]的定义是: “关于数据的数据”。即元数据是描述流程、信息和对象的数据。这些描述涉及像技术属性(例如, 结构和行为)这样的特征、业务定义(包括字典和分类法)以及操作特征(如活动指标和使用历史)。由于元数据可以确保业务系统数据的准确性、一致性和完整性, 同时也通过数据组合和集成形成各类复杂的系统信息。提供完整的应用系统数据现状视图。元数据模型一般运用在数据仓库应用系统中^[3]。但是也可以应用到其他领域。比如图书馆的管理性元数据^[4]就是一种业务的应用。有些学者还把元数据进行分层管理并应用到一般的信息化管理系统中^[5,6]。

对于日志记录的主数据可以采用元数据来进行描述。但是对于一些动作操作也可以采用元数据来描述, 这就引入了元数据对象。所谓元数据对象, 一方面包括了对象属性, 即元数据描述, 另一方面还包括对象方法, 也即是元数据操作。对于日志记录可以把它封装为一个业务对象。日志记录对象包括多个属性。有的属性是包容运行日志记录的, 有的是运行日志记录的子属性。可以把日志记录属性做一个标准模板进行配置, 按照元数据配置的日志记录标准模板可以根据用户要求进行调整和扩充。虽然日志记录对象的属性多种多样, 但可以归纳为三类, 一类是基础属性, 一类是通用属性, 一类是专用属性。每种属性都是一类业务对象。

基础属性说明给属性是所有运行日志记录都必须具有的属性。通用属性是一些运行日志记录比较常用的和通用的属性。专用属性是一些特殊的运行日志记录所具有的特别属性。基础属性包括类别属性、日志

属性、位置属性、组织属性、时间属性。通用属性包括操作对象属性、操作者属性、操作动作属性。

类别属性描述日志记录的日志记录类别, 也就是事件分类。日志属性表示日志记录归属哪一项日志对象。位置属性描述发生日志记录的地点, 组织属性描述日志记录归属的组织单位, 时间属性表示日志记录的时间。操作对象属性描述日志记录中要操作的对象, 如设备对象、物资对象、人员对象等。操作者属性说明日志记录是由谁启动的操作。操作动作属性表示操作者对被操作对象施加了什么动作或者采取了什么行为。一类是动作属性, 一类是类别属性、一类是权限属性、一类是时间属性、一类是位置属性、一类是组织属性等。

这就要求元数据对象模型有三个方面的内容: ① 负责存储和维护元数据库中的元数据; ② 负责检索工具、数据访问接口模块、接收查询结果模块等之间的消息传递。③ 负责协调各模块和工具之间的工作, 分析数据之间的关系, 调度数据之间的活动, 实现对象关系映射。

2.2 运行记录的数据模型

可以用关系数据库模型来描述元数据模型之间的关系。基于元数据的运行记录的数据模型包括元数据配置模型和运行模型两部分。配置模型描述的是运行记录元数据模板的组成、权限定义、规则定义、策略定义、控制定义等内容。运行模型则是基于配置信息产生记录本。通过数据模型, 可以方便地描述关键业务的业务规则、活动的依赖关系以及操作的处理等特征。它们都通过统一的关系结构来定义。图1给出了基于关系结构的元数据运行日志记录的数据模型的ER图(限于篇幅只给出核心表结构)。

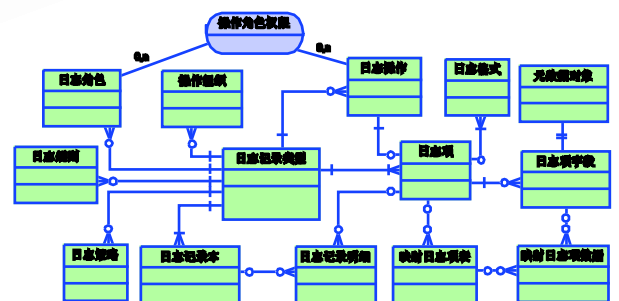


图1 元数据运行记录的关系数据模型

在配置数据模型中, 日志记录类型是最基本的出发点。在电力行业中, 这种日志记录包括调度日志记录、变电站日志记录、通信日志记录、班组日志记录等。

日志项也就是任务项，每种日志类别都有不同的工作任务。如对于变电日志管理中日志项就包括发现设备缺陷登记、设备测温记录、设备巡视记录等。在电力行业中有专门的规范来制定这些工作任务。

日志操作，也就是如何来处理这些工作。这是为了实现权限而派生出来的数据结构。在系统中一般表现为日志菜单。日志菜单与日志角色形成联合，可以根据不同的角色产生不同的日志菜单，也就拥有不同的操作权限。一个日志操作针对一个日志项，是一种强约束的一对一关系。

日志项字段是日志项的具体内容。比如设备缺陷登记项就包括缺陷发现人、设备名称、发现时间、发现位置、缺陷级别、缺陷内容等。这些字段就可以映射到元数据对象，这些对象包括标准的数据对象(如字符对象、时间对象、日期对象、大文本对象)和非标准对象(如部门对象、人员对象、设备对象、物资对象、位置对象)。日志记录项的输入格式和展现格式也有很多种，如下拉列表、选择列表、时间选择、人员选择等等，这些都是通过元数据对象来描述。这些记录从业务上可以分为两大类，一类是事件记录，一类是操作记录。事件记录都是有时间、事件源、对象、要进行统计、汇总和分析。操作记录同样包括时间、操作者、操作对象、这个要有关联性，这个操作可能是应付一个对象，而且这个操作对象还有状态。

日志项对于日志项字段是一对多关系，这就有一个排序和显示的问题。由于日志项、日志项字段多保存数字化的数据，而展现的格式是通过文本或者记录展现出来。这也就要求有一个专门负责装饰的日志格式。日志格式设置主要是通过配置一些解释性和连接性的文字把日志项字段组合成一句话或者一段话，甚至包括字段前后的顺序和标点符号。通过日志格式把存储在关系数据库的数据对象转化为非关系数据的文本语言。

日志规则是实现日志项之间的内部联系而采用的通用规定说明。比如在一个设备缺陷处理过程中，要涉及到设备缺陷登记、设备缺陷处理、设备缺陷验收。由于我们是定制了这三个日志项，他们之间互相并没有关系。所以在日志规则中就会定义一个缺陷处理规则，用这三个日志项中都有的设备名称日志项字段把这三个日志项联系起来，形成一个统一的业务处理流程。

日志策略主要是针对不同的日志类别有不同的算法实现和权限规则。

2.3 元数据对象模型

上述的关系数据模型只是定义了数据的平台的业

务。最后都是为后台的业务服务。这些业务就是把元数据规则的录入界面展现给有相应权限的用户，当用户进行了数据录入后再通过元数据规则转化到数据库中保存。最后基于元数据形成的数据还要实现分析、抽取、统计、汇总等功能。

首先要定义记录的元数据。所谓记录的元数据，就是各个记录的记录项类别。其结构如图2所示。记录项类别主要包括通用字符、数字、日期、大文本、特殊字符(如选择部门、人员、选择设备、物资等等)。对象之间还有父子关系。形成一个树结构。

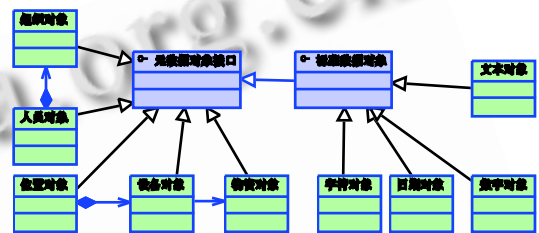


图2 元数据对象结构类图

元数据对象接口是祖先接口，标准数据对象接口继承元数据对象接口。而一些常用的数据类型，如字符、日期、数字、文本等都实现标准数据对象接口。

组织对象是对组织的元数据对象描述，组织本身是一个树状结构。人员对象是关联组织对象的元数据对象。人员本身是一个列表对象，但是放在组织里就是一种组合关系。同理，位置对象也是一个树形的元数据对象，设备对象本身是一个列表，但是放到位置对象中就构成了一个树形结构。设备对象与位置对象也构成组合关系。设备对象同时也是一种物资对象，所以，设备对象与物资对象是一种弱的关联关系。

2.4 业务实现过程

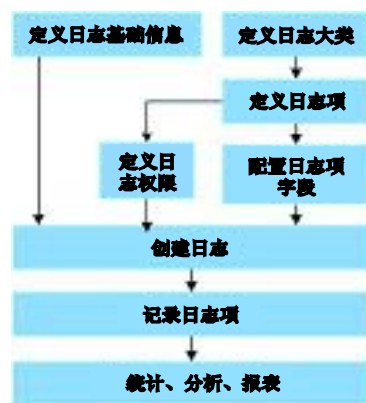


图3 日志系统的业务实现步骤图

基于上述的系统分析，元数据模型的日志系统的业务实现步骤如图 3 所示。

日志系统的步骤总共有 8 步，分别如下：

(1) 首先是定义日志基础信息，定义日志基础信息包括组织对象定义、部门对象定义、班组对象定义、设备对象定义、操作对象定义、日志状态定义等。

(2) 定义日志大类。这样可以保证支持各种类别不同的日志管理。

(3) 定义日志项，这样可以扩展支持多种多样的日志。

(4) 配置日志项字段。就是按照元数据格式来说明各个日志项的属性。

(5) 定义日志权限。权限可以基于日志来赋权，也可以基于日志项来赋权。

(6) 创建日志。根据上述的日志配置信息可以创建一个日志。

(7) 新增、修改和删除日志项。根据创建的日志和上述的日志项配置信息可以动态地创建日志项。

(8) 对日志项的统计、分析和生产报表。由于具体的日志项是最原始的数据源，所以可以按照元数据配置信息和数据生成规则，多维地分析、统计。并能按照各种既定的报表规则生成报表。

3 元数据业务引擎实现

元数据引擎组件将配置模型和运行模型有机地结合在一起，它根据其中定义的业务规则对业务过程中的各项业务活动的解释以及数据转移等工作进行控制和协调。元数据引擎组件是通用运行记录平台的控制中心。

3.1 应用框架

基于元对象模型动态运行记录系统主要是三个部分：第一是配置管理；第二是客户端处理管理；第三是业务逻辑引擎组件管理。如图 4 所示。

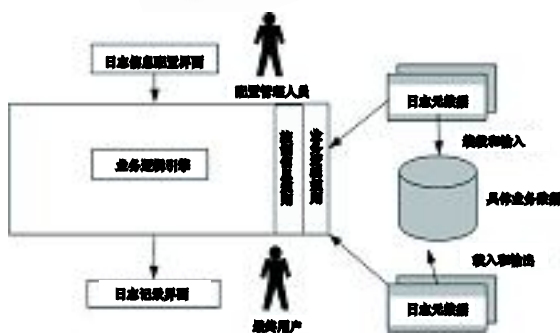


图 4 应用框架

客户端业务部分主要是基于配置信息和配置规则。配置部分主要基础信息的配置、日志类型定义、日志定义、日志项定义、日志项字段定义等。基础配置信息在进入引擎层的时候全部都转化为相关的元数据对象。

业务逻辑引擎部分是联系配置部分和客户端业务部分的中枢。业务逻辑引擎对配置层生成的元数据对象进行分析、解释、组合、分离、合并等操作，由业务逻辑引擎组件生成录入和处理界面，提供给最终用户进行录入、修改和查询。一方面它把配置信息生成录入和处理界面提供给用户进行增加、修改和删除操作，并且把用户提交的录入和修改数据转换成元数据格式保存到关系数据库中，另一方面，当用户进行查询、统计、分析和生成报表的时候，业务逻辑引擎组件把数据库的元数据进行组合、集成、提取等业务规则生成各种用户需要的查询、分析结果和报表。

3.2 逻辑引擎架构

引擎架构主要分为三层：分别为表现层引擎层、逻辑引擎层和信息配置适配器层。如图 5 所示。

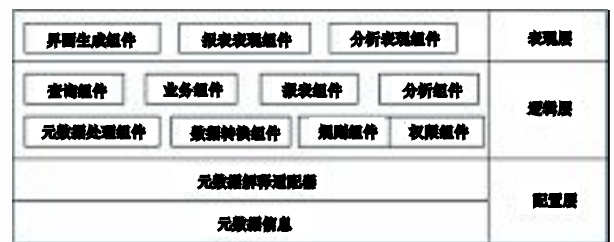


图 5 逻辑引擎架构图

表现层组件主要是生成用户的表现界面，包括用户录入的编辑界面，用户查询和分析的图形和列表界面和用户生成的报表界面。表现层组件有界面生成组件、报表表现组件和分析表现组件。界面生成组件负责按照日志格式、日志项字段生成用户可以录入的表现层界面。可以针对不同的用户设定自己的输入输出界面风格。可使用满足用户个性化输入输出的元数据，诸如背景图片、背景音乐、颜色选择元数据的定义等。通过展现元数据的控制，使客户端的数据按照业务展现元数据的描述，灵活、动态并且按照用户的要求实现个性化显示。

逻辑引擎层由多个组件构成。分别为元数据处理组件、数据转换组件、规则库分析组件、权限组件、查询组件、业务组件、报表组件和分析组件。

元数据处理组件实现把用户录入数据通过规则生成 CRUD 操作，保存到关系数据库中。存储方式有两

种，一种是按照业务表保存，即一个日志项对应一个业务表，这样可以满足复杂的 SQL 查询。还有一种就是直接保存元数据，类似于数据仓库的事实表。这种保存模式会造成事实表的数据巨大，但可以方便进行组合统计、多维分析和数据萃取。

数据转换组件主要是把用户录入的数据按照规则进行转化成业务的元数据，结合元数据处理组件把数据保存到数据库中，也实现把用户提交的查询转化通过元数据转化为用户可以理解的显示格式。

规则库分析组件，基于用户建立的规则，实现元数据在用户提交的数据和关系数据之间的映射方式进行解释和定义。当出现复杂的业务流程要涉及到多个日志项的时候，通过日志项字段的关联性，把多个日志项组合起来完成某一种特定的工作任务。这是一个简化版本的动态建模组件。

权限组件主要是控制和管理用户操作的权限。通过使用角色管理元数据分别对角色、日志项、日志项字段对象等可控对象进行管理，通过角色配置中对这些元数据进行集成、融合。从而既可以处理千变万化的权限需求，又可以灵活有效地控制权限管理粒度。最终达到业务和安全技术相融合。

查询组件，在数据转换组件和规则库分析组件的支持下，用于用户查询语句生成的业务逻辑组件。报表组件和分析组件与查询组件类似，都是在数据转换组件和规则库分析组件的支持下，生成用户的报表和分析内容。

信息配置层主要是有元数据和元数据解释适配器组成。配置元数据信息可以是关系数据库，也可以是 XML 文件格式。通过元数据解释适配器来耦合多种元数据来源。

4 具体应用

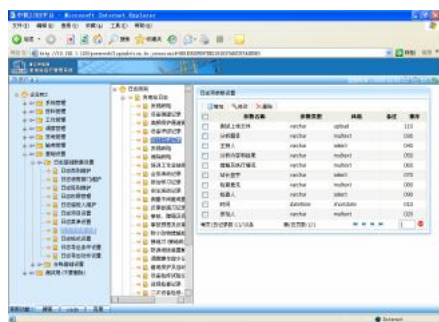


图 6 日志项字段配置界面

本平台在具体应用上对数据库和应用服务器都没有限制。一般情况下，可采用关系数据库作为后台数据库。由于元数据对象模型也是基于对象实现，可采用 OO 语言 java 来实现。

实际应用中采用基于 B/S 模式的 MVC 架构框架。中间件服务器采用支持 EJB 的 J2EE 应用服务器，采用 Oracle 10g 作为后台数据库系统，图 6 是平台中日志项字段的配置界面。

5 结语

本文通过设计一种元数据对象模型，解决复杂多变的运行日志处理系统，该平台一方面采用元数据对象实现业务与界面的紧密结合，另一方面基于元数据对象把有价值的业务数据通过关系数据库保存起来。在这些过程中既不必关心系统结的等关联关系，也不必在程序中固化这些关系，其应用特点如下：

(1) 采用动态配置，可以实现灵活的业务集成和延伸，提高了平台的可扩展性和可伸缩性。

(2) 根据用户的不同数据访问权限定制出不同的维护界面，权限细化到字段信息维护，保证了数据的安全。

(3) 元模型能够对自身进行描述，因此是最稳定的数据，能够实现对系统所有表、字段、约束等管理，保证了数据维护的灵活性。

(4) 系统具有良好的包容性，为用户提供一个统一的业务处理和配置界面，为开发人员提供一个统一的数据访问接口，既方便了使用者又方便了开发者。这样就保持各个业务系统多样性自治性，又可以保证数据的一致性。

(5) 系统的元模型的可以维护到不同的应用系统的不同的数据库表结构，并不影响程序运行，增强了系统的健壮性和模型的可移植性。

(6) 由于采用元数据生成界面，可以快速搭建应用原型，便于开发人员与客户交流，降低了理解上的差异，缩短了项目开发周期。

目前该平台模型已在广东湛江供电局生产管理系统中的变电、输电和调度运行管理项目中进行了实践，取得了较好的效果。而且，由于具备了元数据模型的自适应模式，本平台可以应用推广到其他任何有运行日志记录的领域和行业中去。

(下转第 5 页)

(上接第 10 页)

参考文献

- 1 张晓林.元数据研究与应用.北京:北京图书馆出版社, 2002.
- 2 彭蓉,刘进,何璐璐,刘超.公共仓库元模型:数据仓库集成标准导论企业数据仓库规划,建立与实现.北京:机械工业出版社, 2004.
- 3 杨鸿宾,宋明.元数据管理平台总体架构设计研究.计算机系统应用, 2007,16(11):17-20.
- 4 郑巧英,王绍平,孙华,黄镛,汤莉华.数字图书馆中基础管理性元数据框架研究.图书馆杂志, 2008,27(6):56-62.
- 5 马良荔,汪厚祥,李永杰.构件分层元数据对象参考模型的设计.计算机应用研究, 2007,24(10):282-285.
- 6 袁满,陈永恒.一种新型的面向信息化应用的数据元支撑元模型.计算机应用研究, 2008,25(7):2075-2077.
- 7 朱建勇,赵振南.基于面向对象数据访问中间件设计与应用.长江大学学报(理工卷), 2006,(5):658-660.
- 8 穆殿宝.基于用户-权限的通用数据维护元模型的设计.红水河, 2008,5(1):50-52.