

## 基于 openEHR 的临床医疗数据仓库构建方法

王冬冬<sup>1</sup> 叶 枫<sup>1</sup> 吕旭东<sup>2\*</sup> 闵令通<sup>2</sup> 王 利<sup>2</sup>

<sup>1</sup>(浙江工业大学经贸管理学院, 杭州 310023)

<sup>2</sup>(浙江大学生物医学工程与仪器科学学院, 杭州 310027)

**摘 要:** 医疗信息的复杂性和动态性给医疗信息系统带来巨大的挑战, openEHR 规范的两层建模思想提高医疗信息系统的灵活性, 适应医疗信息需求的变化。但面临查询数据量巨大、查询条件复杂多变的个性化操作需求时, 基于 openEHR 两层建模方法实现医疗信息系统查询性能较低, 这主要与底层数据存储模型有关。数据仓库多维数据模型具有高性能查询的优点, 通过建立满足个性化需求的数据集市, 可以加快数据查询速度。由于基于 openEHR 规范两层建模方法实现的医疗信息系统的特殊性, 传统的数据仓库构建方法使得用户的参与性较低并且费时费力。针对这一问题, 以 openEHR 规范的两层建模思想为基础, 提出模板到动态数据仓库多维数据模型的映射方法, 实现多维数据模型的用户可配置, 利用映射路径加快 ETL 的实施, 为医疗行业提供一种可操控、可扩展并真实反映用户需求的数据仓库构建方法。对该方法进行性能上的验证, 结果表明, 在 10 861 522 条数据中查询 329 条数据时, 基于该方法构建的数据集市是基于 openEHR 两层建模方法生成的数据库的 5.6 倍, 是基于传统方法构建的数据集市的 0.97 倍。

**关键词:** openEHR; 模板; 数据仓库; 多维数据模型

中图分类号 R318 文献标志码 A 文章编号 0258-8021(2016) 02-0162-07

## Construction Method of Clinical Medical Data Warehouse Based on openEHR

Wang Dongdong<sup>1</sup> Ye Feng<sup>1</sup> Lv Xudong<sup>2\*</sup> Min Lingtong<sup>2</sup> Wang Li<sup>2</sup>

<sup>1</sup>(College of Business Administration, Zhejiang University of Technology, Hangzhou 310023, China)

<sup>2</sup>(College of Biomedical Engineering and Instrument Science, Zhejiang University, Hangzhou 310027, China)

**Abstract:** Complexity and continuous evolution of healthcare information is the biggest challenge for healthcare information systems, and two-level information model from openEHR can facilitate healthcare information systems to adapt to the change of healthcare information requirements. However, the medical information system based on openEHR two-level modeling displays low performance when facing with specific operational demands such as a large number of data query and complex and mutative query conditions. This is mainly related to the fundamental data storage model. The multidimensional data model of data warehouse has the advantage of high-performance query, the data mart that is established for fulfilling users' particular demand can increase the particular query speed. Due to the particularity of the medical information system that based on openEHR two-level modeling, the outcomes for traditional data warehouse construction method include low users' participation and time consuming. In this work, a mapping method between template and multidimensional data model of data warehouse based on openEHR two-level modeling was proposed to solve this problem, aiming to achieve the configuration of multidimensional data model by users, and use the mapping path to accelerate the implementation of ETL. This research provided one kind of dirigible and scalable data warehouse construction method reflecting user's requirements for medical industry. We also validated this method in performance. Results showed that the data mart that was built based on this method was 5.6 times faster than the database

which was generated based on openEHR two-level modeling and was 0.97 times than the data mart that was built by traditional method when querying 329 results in 10861522 records.

**Key words:**openEHR; template; data warehouse; multidimensional data model

引言

医疗领域的医学知识和医疗业务需求复杂多变,采取传统的医疗信息系统开发方法,工程技术人员不仅要在医疗信息系统开发之时即与医疗领域专家反复讨论确认,而且在系统的使用过程中还要不断对系统进行维护和更新,这使得医疗信息系统很难及时响应医疗人员对医疗信息的需求。openEHR 规范作为一套开放的 EHR ( electronic health record) 体系结构,其核心在于将医疗领域知识从具体的临床信息中分离出来,并建立两层模型——参考模型和原型模型<sup>[1-2]</sup>。按照 openEHR 规范的设计,系统底层结构的开发主要基于参考模型,而系统中所涉及的领域概念全部由原型定义,这就有效降低了系统底层结构对领域知识的依赖性,从而使系统对领域知识变化的适应能力大大增强。领域知识层由原型和模板组成,允许领域专家直接定义,从而“把临床医生放回了驾驶员的座位上”<sup>[3]</sup>。

目前,国际上开展了很多基于 openEHR 规范的医疗信息系统的研究<sup>[4-5]</sup>,对 openEHR 两层建模思想进行了验证。在基于 openEHR 规范构建的医疗信息系统数据持久化层方面,国内外已研究出基于 openEHR 两层模型方法建立的医疗数据存储平台。该平台通过读取由领域专家定义的 openEHR 领域模型来满足相应的数据存储需求,实现医疗数据存储结构的高度可适应性和可扩展性。其中,开源的基于 openEHR 体系结构的 Opereffa 原型系统底层使用关系数据库存储医疗数据<sup>[6]</sup>,瑞典林雪平大学的 Sergio Miranda Freire 等实现了使用 XML 数据库存储基于 openEHR 规范的医疗数据文档<sup>[7]</sup>,日本会津大学的 Aastha Madaan 等实现了由原型驱动的基于 NoSQL 数据库的医疗数据存储平台<sup>[8]</sup>。除此之外,浙江大学的王利等为了管理复杂和不断发展的医疗信息,利用 openEHR 两层模型和原型驱动方法的高度可适应性和可扩展性,设计并实现了一个由原型驱动的基于关系型数据库的医疗数据存储平台,由原型映射到关系型数据库表( ARM, archetype relational mapping) 来实现灵活的数据存储。在 ARM 的基础上,该研究将 AQL 扩展为具有完整数

据操作功能的数据操作语言,并在数据存储平台上进行了技术实现,从而使该数据存储平台通过一组网络服务对外提供灵活的数据访问<sup>[9]</sup>。

基于 openEHR 两层建模实现的医疗信息系统提高了系统的灵活性,适应了医疗信息的变化和更新,但面临数据查询量巨大、查询条件复杂多变的个性化操作需求( 报表、集成视图等) 时,基于 openEHR 两层建模实现的医疗信息系统查询性能较低。这主要与底层数据存储模型有关,数据仓库多维数据模型拥有高性能查询的优点,通过改变基于 openEHR 两层建模实现的医疗信息系统底层数据存储结构,可以解决个性化操作需求中存在的性能问题。对于此类问题,目前采用数据驱动和应用驱动相结合的数据仓库开发方法,即根据数据源和具体的需求来设计多维数据模型,再进行相应的 ETL( extract, transform, load) 设计实现,建立数据仓库<sup>[10]</sup>。由于基于 openEHR 规范两层建模方法实现的医疗系统的特殊性,系统持久化层会伴随领域知识的改变而发生变化,相应的数据仓库多维数据模型以及 ETL 设计都需随之发生改变;采用传统的数据仓库构建方法,在数据仓库的开发与维护过程中,技术开发人员需要不断地与用户进行沟通交流来明确用户的需求,费时费力且容易出错,用户的参与性较低,不能很好地适应领域知识的变化。

针对这一问题,为了提高数据仓库构建时用户的参与性,加快数据仓库的构建速度,更好地满足用户的个性化需求,本研究以国际标准 openEHR 为基础,采用两层建模的方法,提高用户的可操控性,利用数据仓库多维数据模型高性能查询的优点,提出模板到动态数据仓库多维数据模型的映射方法。与传统的数据仓库构建方法相比,该方法实现了数据仓库多维数据模型的用户配置,提高了建立数据仓库时用户的参与性;在多维数据模型与数据源之间建立了映射关系,根据自动生成的映射路径可以快速地访问数据源中的数据结构、数据类型等信息,加快了 ETL 工作的进行,降低了构建数据仓库的周期。

1 多维数据模型映射方法

openEHR 原型通常是对临床知识完整定义,而

在实际的医疗环境中,往往不需要记录如此完整详尽的信息。openEHR 模板是根据实际临床需求,对若干原型合理组合并做进一步的语义约束,不会对原型施加任何新的语义。模板与原型的不同之处主要在于:原型通常需要在国家甚至国际层面上进行统一制定,以保证得到的原型具有通用性,可以被广泛共享。而模板的制作通常是相对局部化、本地化的事情,每个软件系统都可以根据自己的本地化需求,定义出一系列特定的模板。研究模板到动态数据仓库数据存储模型映射方法,更能切实地满足医疗工作者的个性化需求<sup>[11-12]</sup>。

基于 openEHR 原型关系映射方法构建医疗信息系统,在显著提高数据持久化性能的同时,使医疗信息系统能够适应医疗信息需求快速发展变化的形势,促进医疗信息系统的应用<sup>[9]</sup>。目前,该构建方法被广泛采用,本研究选取基于 openEHR 原型关系映射方法生成的关系型数据库,作为构建数据仓库的数据源。

数据仓库技术采用多维数据模型,多维数据模型把数据看作是多维空间中的点集,把集合的属性分为维和度量两类:维属性用来描述度量属性,是多维空间的维度;度量属性的值用来进行分析处理,是多维空间中的点。根据事实表和维表的关系,又可将常见的模型分为星型模型和雪花模型。两者在数据冗余度和查询性能上有所区别:雪花模型冗余度低,查询性能较低;星型模型虽然冗余度高,但查询性能好。因此,在冗余可以接受的前提下,实际中运用星型模型更多,也更有效率<sup>[13-14]</sup>。

采用基于 openEHR 的临床医疗数据仓库构建方法,其主要技术路线如图 1 所示。

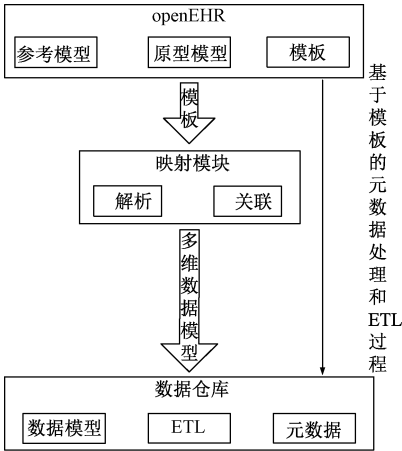


图 1 技术路线

Fig. 1 Technology route

模板真实反映了基于 openEHR 规范两层建模方法实现的医疗信息系统底层数据存储结构。该技术路线从模板出发,通过映射模块在模板与多维数据模型之间建立关联,在此基础上进行基于模板的元数据处理和 ETL 过程,实现数据集市的建立。映射模块实现模板到动态数据仓库多维数据模型的映射,具体实现方法为:映射模块对传入的模板进行解析,生成反应模板内部数据结构信息的树形结构,根据原型关系映射规则对树形结构中的可选项进行筛选。用户通过对树形结构中节点的挑选,组合成满足需求的多维数据模型,并自动收集模板到多维数据模型的映射关系以及必要的元数据。

ASP.NET 是建立、部署以及执行 Web 应用程序的平台,它为构建新一代动态网站和基于网络的分布式应用提供了有力的支持。ASP.NET 下 MVC 设计模式将程序分成相对独立而又能协同工作的 3 个部分——模型、视图、控制器,可以清楚地分离用户界面与业务逻辑,能够为系统开发提供基本的分析方法和清晰的设计框架<sup>[15]</sup>。本研究基于 ASP.NET 下 MVC 设计模式开发来实现。

1.1 模板解析

模板遵循 TOM(template object model)定义,原型由 ADL(archetype definition language)定义。openEHR 模板是根据实际需求对若干原型的合理组合,在模板解析中需要访问模板引用的原型。openEHR 参考模型和原型定义了大量的类和属性,以及类之间的关系,尤其是参考模型,它不仅定义了 openEHR 的上层组织结构,而且对 openEHR 中各类信息的组织结构以及数据类型、数据结构等都进行了详细定义。因此,直接访问完整的参考模型和原型的实施将会是一项非常庞大的工程。目前,国际上已经出现了多个 openEHR 规范实施项目,其中比较有代表性的是瑞典 Linköping 大学 Rong Chen 等人组织开展的 openEHR JAVA 开源参考实施项目。该项目已经对 RM 和 AM 进行了比较完整的实现,因此,本研究 RM 和 AM 的实现直接基于此 JAVA 参考实施项目。根据实际需要,采用了 openEHR JAVA 的模块 adl-parser 和模块 xml-serializer。模块 adl-parser 实现了 ADL 文件到 AOM/AP 对象的解析,模块 xml-serializer 实现了 AOM/AP 对象到 XML 文件的序列化。通过以上两个模块,可以将 ADL 文件转化为既方便计算机处理又具有一定可读性的 XML 数据文件。JAVA 类文件并不能直接在 ASP.NET 下运行,IKVM.NET 是开



源的基于 .NET CLR 的 JAVA 虚拟机,该项目虽然在安全性上有待改进,但实现了 JAVA 程序在 .NET 下的运行。本研究通过引用以上两个模块和 IKVM.NET,将 ADL 文件转化为方便计算机处理并且具有一定可读性的 XML 数据文件。

JSON 作为一种轻量级的数据传输格式,可以在多种语言之间进行数据交换。JSON 易于阅读和编码,并且是 JavaScript 规范的子集,能够被支持 JavaScript 的浏览器所解析;相比 XML,它减少了解析时带来的性能和兼容性问题,使其成为了理想的数据交换语言<sup>[16]</sup>。因此,本研究将 JSON 作为数据交换语言。在实现中,主要是 XML 文件到 JSON 对象的转换,这一过程借助 XML2JSON 插件完成。

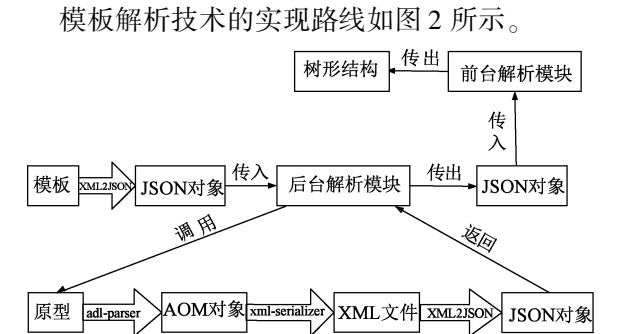


图 2 模板解析路线

Fig. 2 Template parsing route

将模板转化为 JSON 对象后,传入后台解析模块,后台解析模块依次调用需要的原型,原型经过一系列的转化后转变为 JSON 对象,返回后台解析模块,后台解析模块解析模板,并生成一个真实反映模板内部结构的 JSON 对象,传入前台解析模块,前台解析模块遍历该 JSON 对象,生成模板对应的树形结构。

在本研究中,将基于 openEHR 原型关系映射方法生成的关系型数据库作为构建数据仓库的数据源。为了真实地反映数据源中的数据结构信息,根据原型关系映射规则,对树形结构中的可选项进行了筛选,原型关系映射规则<sup>[9]</sup>如下:

- 1) 每个原型映射为一个关系数据库表,数据库表名为原型名。
- 2) 每个基本类型的原型属性映射为一个关系数据库表字段,字段名为属性名,字段类型为属性数据类型,字段长度为属性数据长度。
- 3) 每个集合类型的原型属性映射为一个单独的数据库表,数据库表名为“原型名\_集合类型属性名”,包括主键字段,字段名为“原型名\_集合类型属

性名”;关联到原型对应的数据库表的外键字段,字段名为原型名,字段类型与原型对应的数据库表的主键字段类型相同;集合类型属性对应的字段,字段名为集合类型属性名,字段类型为集合类型属性数据类型,字段长度为集合类型属性数据长度。

4) 每一个 archetype slot 类型的原型属性映射为一个单独的数据库表,数据库表名为“原型名\_archetype slot 属性名”,包括主键字段,字段名为“原型名\_集合类型属性名”;关联到原型对应的数据库表的外键字段,字段名为原型名,字段类型与原型对应的数据库表主键字段类型相同; archetype slot 类型属性对应的字段,字段名为“archetype slot 类型属性名”,并关联到目标原型对应的数据库表。

根据以上规则,树形结构中能够映射到关系型数据库表字段名的节点为可选项,其他节点为不可选项。

树形结构各节点 name 按照是否可选进行了不同的命名方式。不可选项 name 为对应模板层次结构中节点的属性名,可选项 name 由对应模板层次结构中节点的属性名、属性数据类型、该属性节点在模板中的完整路径三者拼接而成,以方便信息的传递。

1.2 可操控配置

个性化需求复杂多变,而模板包含了满足实际应用的信息,用户通过对解析模板生成的树形结构中有价值节点的选择,组合成满足个性化需求的事实表和维表,构建数据集市的多维数据模型。具体实现方法如下:

根据数据仓库多维数据模型的相关定义,允许用户在建立数据集市时创建一张事实表和多张维表。在创建表的过程中,表名由用户手动输入,并进行唯一性检验,避免表名重复。用户通过对树形结构中有价值节点的选择,将该选中项的 name 信息传送到对应表,生成对应表的行信息,用户可以指定已创建表中的某一行为主键或某些行组合成复合主键,并可以对已创建的事实表和维表进行主/外键关联操作,组合成多维数据模型。

数据集市是聚焦的、面向特定主题的,它通常致力于单一的某个领域,且为特定用户服务。为了方便对数据集市的使用与管理,在用户建立多维数据模型时,需用户输入数据集市的唯一性命名以及面向对象等元数据信息。

1.3 数据库生成

该模块的主要功能是根据用户已经建立好的

多维数据模型,生成数据库,并收集部分元数据信息。

用户输入命名的事实表名和维表名通常不符合实际数据库表名的命名规则,需经过解析转化符合命名标准的表名。解析用户配置生成的事实表和维表的各个行信息,得到该行的属性名、属性数据类型、属性节点的完整路径。解析后的属性名和属性数据类型主要为生成数据库表字段服务,而属性节点的完整路径提供 AQL 查询,可以查询到生成表字段的对应节点在原型中的信息,并可与通过原型关系映射生成的关系数据库中的表字段关联起来,方便数据的集成处理。

解析后的属性名并不能直接作为生成数据库表的字段名,需经过解析和唯一性处理生成对应表的字段名,字段类型为对应属性数据类型。然后,收集主/外键关联信息,生成对应的创建关系型数据库表的 SQL 语句,在数据库中创建事实表和维表,并记录下生成的数据集市名、数据集市面向对象、表名、表字段、表字段对应的属性节点的完整路径等元数据信息,便于对数据集市的管理。

2 结果

为了验证该方法是否可行,笔者针对查询检验信息的相关报表中存在的性能问题,根据模板到动态数据仓库多维数据模型映射方法,构建了满足需求的多维数据模型,通过专业的 ETL 工具 SSIS (Microsoft SQL Server Integration Services),对数据进行了集成处理,建立数据集市,希望通过改变底层数据存储结构,提升查询性能,实现数据仓库的快速构建。

选取了与检验信息相关的模板中有价值的节点,配置基于 openEHR 模板的多维数据模型,生成的数据库表结构如图 3 所示。该模型为包含一张事实表和三张维表的星型模型,涵盖了满足个性化需求的所需数据。其中,事实表 FactLaboratoryTest 存放了检验结果的相关数据,复合主键为 TestReport\_ID、ItemOrder。维表 DimPatient 存放了病人基本信息的相关数据,主键为 PatientIdentifier\_Identifier;维表 DimAdmin\_PatientAdmission 存放了病人就诊的相关数据,主键为 EncounterIdentifier;维表 DimInst\_LaboratoryTestRequest 存放了检验申请的相关数据,主键为 RequestorIdentifier。3 个维表通过主键与事实表 FactLaboratoryTest 建立关联。

在 ETL 过程中,基于 openEHR 原型关系映射生

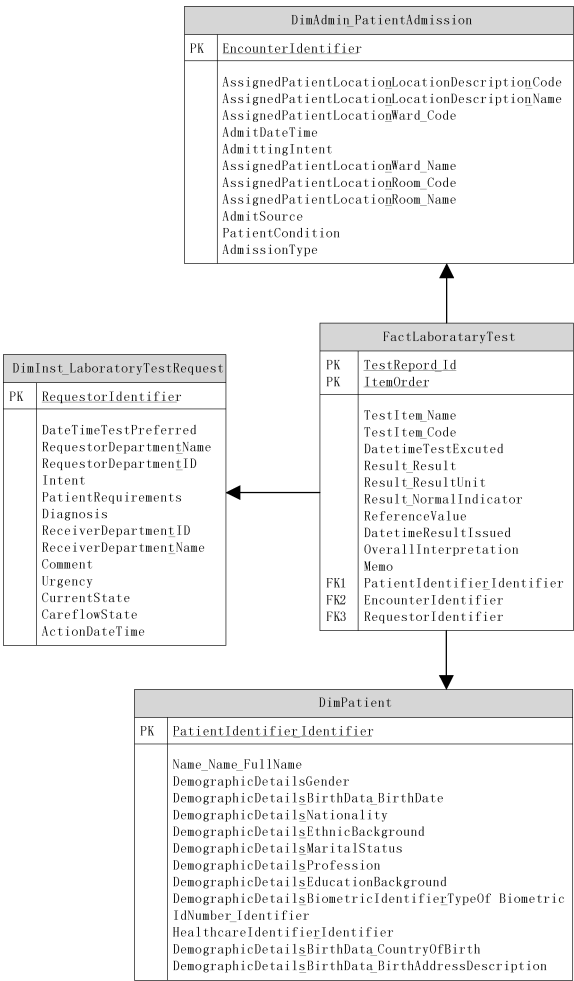


图 3 数据模型  
Fig.3 Data model

成的关系型数据库为数据源,利用构建多维数据模型时收集的表字段的完整路径与基于 openEHR 原型关系映射生成的关系型数据库中的表字段进行关联,快速地查找到其在数据源中对应数据的数据结构、数据类型等相关信息,在此基础上进行数据的集成处理。数据抽取频率设定为每天一次,抽取策略采用全量抽取,在 ETL 引擎中完成数据转换,然后对数据进行装载。

为了检验查询性能是否得到了提升,在基于模板到动态数据仓库多维数据模型映射方法建立的数据集市和基于传统方法建立的数据集市、基于原型关系映射生成的关系型数据库间进行查询性能对比,挑选了三者中的部分数据作为测试数据集,数据包括57 465位病人基本信息、1 210 798条检验申请记录、840 746条检验报告记录、10 861 522条检验结果记录。分别在三者中对某病人的相关信息、26 条检验申请记录、24 条检验报告记录、329 条检验结果记录进行了查询。测试电脑配置 CPU 为

Intel(R) Core(TM)2 Duo CPU,内存为 4GB,操作系统为 Windows 7.1 Enterprise 32 位,数据库为 SQL Server 2012 Enterprise 32 位。在三者中,分别进行了 3 次查询,所花费的时间以及各自平均查询时间的结果如表 1 所示。

表 1 数据查询时间  
Tab.1 data query time

数据库	T <sub>1</sub> /ms	T <sub>2</sub> /ms	T <sub>3</sub> /ms	平均时间/ ms
基于模板到动态数据仓库多维数据模型映射方法构建的数据集市	276	261	287	274
基于传统方法构建的数据集市	268	249	279	265
基于原型关系映射生成的关系型数据库	1 515	1 452	1 593	1 520

查询结果表明,在基于模板到动态数据仓库多维数据模型映射方法建立的数据集市中,对某病人的检验相关信息进行 3 次查询所花费的时间均值为 274 ms。在基于传统方法建立的数据集市中,花费的时间均值为 265 ms;在基于 openEHR 原型关系映射生成的关系型数据库中,花费的时间均值为 1 520 ms。在查询性能上,基于模板到动态数据仓库多维数据模型映射方法是基于传统方法建立的数据集市的 0.97 倍,基本持平。与基于 openEHR 原型关系映射生成的关系型数据库相比,基于模板到动态数据仓库多维数据模型映射方法构建的数据集市在性能上提升了约 5.6 倍。

总体表明,根据模板到动态数据仓库多维数据模型映射方法,可以配置满足个性化需求的多维数据模型,利用多维数据模型与数据源中数据建立的映射关系,方便了 ETL 过程的进行,实现了数据集市的快速建立。查询时间对比表明,该方法切实可行,大幅提升了个性化需求的查询性能。

3 讨论

在本研究中,基于 openEHR 规范,提出了模板到动态数据仓库多维数据模型的映射方法。与传统的数据库构建方法相比,该方法充分利用了 openEHR 两层建模思想的模板领域知识驱动、易扩展、反映本地化需求等优点,使用户在无需了解数据源底层数据存储结构的基础上,实现了数据库多维数据模型的用户可配置,提高了在数据库构建过程中用户的参与性,避免了数据库开发前期

的需求分析工作,满足了用户的个性化需求。在数据库多维数据模型的构建过程中,自动收集了数据源与多维数据模型间的映射关系,利用映射路径,方便 ETL 过程的处理,提高了数据库的构建效率。

用户参与数据仓库中多维数据模型的设计有利也有弊。用户临床医疗知识和计算机操作水平的差异,会影响到多维数据模型的质量,并直接关系到数据集市的的服务水平。在这一点上,本研究提出的模板到动态数据仓库多维数据模型的映射方法做了妥协。用户需要在技术人员的指导下进行模型的设计,借助专业的 ETL 工具实现数据集成。

提高用户在临床医疗数据仓库系统开发过程中的参与性,让用户成为数据库构建的主角,是缩短系统开发周期、满足用户个性化需求的重要手段。但是,让用户独立配置数据库还是具有一定的风险,所以领域专家的指导和培训非常重要。通过培训,让用户了解临床医疗数据仓库的相关知识,以及将医疗数据仓库的领域概念与医疗领域知识相结合,对于数据库配置过程中以用户为主体具有一定的可操作性。

4 结论

在本研究中,针对基于 openEHR 两层建模思想实现的医疗信息系统在个性化操作需求时存在的性能问题,提出了模板到动态数据仓库多维数据模型的映射方法。对该方法进行了验证,表明通过该方法能够建立满足需求的多维数据模型,实现多维数据模型的用户可配置。通过对数据的集成,在查询性能上进行了验证,表明基于该方法建立的数据集市与基于原型关系映射方法生成的数据库相比,能够大幅地提升查询性能,与基于传统方法构建的数据集市在查询性能上基本持平。该方法充分利用了 openEHR 规范领域知识驱动的优点,为医疗行业提供一种可操控、可扩展、能真实反映用户需求的数据仓库构建方法。

参考文献

[ 1 ] 曾蕾,姚志洪,刘雷. 双模型健康档案标准 openEHR[J]. 中国医疗设备,2010,25(3):7-10.  
[ 2 ] Beale T, Heard S, Kalra D, et al. OpenEHR architecture overview[J]. The OpenEHR Foundation, 2006.  
[ 3 ] Kalra D, Beale T, Heard S. The openEHR foundation[J]. Studies in health technology and informatics, 2005,115: 153-173.

- [ 4 ] Garde S, Hovenga E, Buck J, et al. Expressing clinical data sets with openEHR archetypes: a solid basis for ubiquitous computing [J]. *Int J Med Inform*, 2007, 76(3):334 – 341.
- [ 5 ] Spath MB, Grimson J. Applying the archetype approach to the database of a biobank information management system [J]. *Int J Med Inform*, 2011, 80(3):205 – 226.
- [ 6 ] Arikan S, Shannon T, Ingram D. Opereffa 2009 [EB/OL]. <http://opereffa.chime.ucl.ac.uk/introduction.jsf>, 2009 – 05 – 14/2015 – 09 – 10.
- [ 7 ] Freire S M. Performance of XML databases for epidemiological queries in archetype-based EHRs [C]//Freire SM, Sundvall E, Karlsson D, eds. *Proceedings Scandinavian Conference on Health Informatics 2012*. Linköping: Linköping University Electronic Press, 2012;51 – 57.
- [ 8 ] Madaan A, Chu W, Daigo Y, et al. Quasi-relational query language interface for persistent standardized EHRs: using NoSQL Databases [M]//*Databases in Networked Information Systems*. Berlin: Springer Berlin Heidelberg, 2013;182 – 196.
- [ 9 ] 王利, 闵灵通, 吕旭东, 等. 基于 openEHR 的原型关系映射方法 [J]. *中国生物医学工程学报*, 2014, 33(4):432 – 437.
- [ 10 ] 何彩升, 邹赛德. 数据仓库技术及其在医院信息管理中的应用 [J]. *计算机医学应用*, 2003, 16(8):418 – 422.
- [ 11 ] Leslie H. International developments in openEHR archetype and template [J]. *HIM J*, 2008, 37(1):38 – 39.
- [ 12 ] 李珍珍. 基于 openEHR 的电子病历系统开发方法研究与实践 [D]. 杭州: 浙江大学, 2008.
- [ 13 ] 李明芳, 刘红, 姚敏. 数据仓库方法论 [J]. *计算机应用与软件*, 2003, 20(1):1 – 2.
- [ 14 ] 宋擒豹, 杨向荣, 沈钧毅. 数据仓库技术研究 [J]. *计算机工程*, 2002, 28(1):125 – 127.
- [ 15 ] 李园, 陈世平. MVC 设计模式在 ASP.NET 平台中的应用 [J]. *计算机工程与设计*, 2009, 30(13):3180 – 3184.
- [ 16 ] 高静, 段会川. JSON 数据传输效率研究 [J]. *计算机工程与设计*, 2011, 32(7):2267 – 2270.