

基于 openEHR 的原型关系映射方法

王 利 闵令通 吕旭东* 段会龙

(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挑战。openEHR 规范的两层建模思想可以提高医疗信息系统的灵活性, 适应医疗信息需求的变化。目前虽然已有采用 openEHR 官方提出的属性 + 路径 (node + path) 和 openEHR 参考模型对象关系映射的 openEHR 数据存储实现方案, 但其性能无法达到临床实际应用的需要。本研究提出一种基于 openEHR 原型和关系数据库的自动映射方法, 将原型映射为关系数据表、原型属性映射为关系数据表字段, 在此基础上设计了原型驱动的数据持久化和数据访问平台。基于原型关系映射方法, 对原型查询语言进行扩展, 添加 INSERT、UPDATE、DELETE 等语法, 使原型查询语言支持完整的数据操作功能。结果表明, 原型关系映射方法与属性 + 路径和 openEHR 参考模型对象关系映射方法相比, 在 1 231 条数据中查询 3 条数据时快 69 倍和 8 倍, 在 128 141 条数据中查询 1 593 条数据时快 2 倍和 17 倍。原型关系映射和原型查询语言等技术不但可以发挥两层建模方法在医疗信息表达、存储和访问等方面的灵活性优势, 而且能大幅提高数据访问的性能。

关键词: 原型关系映射; 原型查询语言; 原型; 关系数据库; openEHR

中图分类号 R318 文献标志码 A 文章编号 0258-8021(2014) 04-0432-06

Archetype Relational Mapping Based on openEHR

WANG Li MIN Ling-Tong LU Xu-Dong* DUAN Hui-Long

(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. Two-level information model from openEHR can facilitate healthcare information systems to adapt to the change of healthcare information requirements. Persistence methods such as Node + Path promoted by openEHR and openEHR reference model object relational mapping (RMORM) cannot meet the performance requirements of clinical daily use. In this paper, we proposed a novel persistence method based on direct mapping between openEHR archetype and relational database by mapping archetype to data table and mapping archetype attribute to data table column, and designed an archetype-driven data persistence and data access platform. Based on the archetype relational mapping (ARM) method, we extended the archetype query language (AQL) by adding INSERT, UPDATE, and DELETE clauses to support full data manipulation. Results showed that, ARM is 69 times faster than Node + Path and 8 times faster than RMORM when querying 3 results in 1 231 records, and ARM is 2 times faster than Node + Path and 17 times faster than RMORM when querying 1 593 results in 128 141 records. With technics such as ARM and AQL, the data platform can achieve high performance while keeping the advantage of openEHR two-level modeling in healthcare information expression, storage, and access.

Key words: archetype relational mapping; archetype query language; archetype; RDB; openEHR

doi: 10.3969/j.issn.0258-8021.2014.04.07

收稿日期: 2014-03-25, 录用日期: 2014-05-27

基金项目: 国家高技术研究发展计划(863计划)(2012AA02A601); 国家科技重大专项(2013ZX03005012)

* 通信作者(Corresponding author) E-mail: lxldxd7366@gmail.com

引言

随着数字化医院建设的飞速发展, 医疗信息技术和医疗信息系统在医院中得到了广泛和深入的应用。医疗业务需要从传统的基于纸张的病历, 系统地迁移到基于医疗信息系统的电子病历^[1]。但是, 医疗领域信息概念繁多、结构复杂、动态变化的特点, 给现有的医疗信息系统的开发和应用带来了巨大的挑战。首先, 医疗领域包含大量的医学信息, 不同专科、不同疾病都有着不同的信息需求, 很难设计一个涵盖各个科室、各种疾病的信息需求的医疗信息系统。其次, 由于医疗信息化建设水平和管理理念不同, 相同的医疗业务在不同的医疗机构、不同的科室具有不同的信息需求, 同样的医疗信息系统往往需要经过大量定制才能在医疗机构中应用。第三, 医学研究不断发展, 医疗管理水平不断提高, 导致新的信息需求不断涌现, 现有的信息需求不断变化。纸张病历可以通过添加或删除表格, 灵活地适应医疗业务的变化。然而, 对于结构单一复杂的医疗信息系统来说, 频繁的修改将会导致其维护成本和难度急剧上升。传统观点认为, 医疗行业采用高度复杂特殊化的信息系统, 使 IT 成本和医生工作负担都急剧增加^[2]。

openEHR 规范是由 openEHR 组织^[3] 制定的一套开放的电子病历规范, 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, 并建立了两层模型——参考模型和原型模型^[4]。参考模型是一组表达医学知识和概念的通用基础数据类型和数据结构, 通过对参考模型添加约束的方式来定义原型, 大部分的原型都是从少数几个固定的参考模型类派生出来的。医疗信息系统的开发主要基于参考模型, 领域概念全部由原型定义, 这就有效地降低了系统对领域知识的依赖, 从而使系统能够适应领域知识的变化。

国际上开展了许多基于 openEHR 规范的医疗信息系统研究^[5-6], 对 openEHR 两层建模思想进行了验证。但是, 在临床应用的过程中, 还存在一些重大挑战, 其中一个是基于 openEHR 规范构建的医疗信息系统的数据库持久化性能问题。openEHR 官方公布了一种属性 + 路径 (node + path) 方法: 在 openEHR 原型中, 每个原型属性 (node) 具有唯一的路径 (path) 标识。利用 EAV (entity attribute value) 表模型, 构建一张包含少量字段的数据库表来存储原型实例数据, 如图 1 所示。Open EHR-Gen

Framework 提出了一种参考模型 (reference model, RM) 对象关系映射 (object relational mapping, ORM) 方法。将参考模型通过对象关系映射, 构建出对应的关系数据库表来存储原型实例数据。这两种方法将全部原型实例数据集中于少量关系数据库表中, 随着数据量的增长, 数据访问性能急剧降低。使用 XML 数据库存储原型数据^[7] 性能与关系型数据库相差较大, 无法满足实际医疗业务的需求。

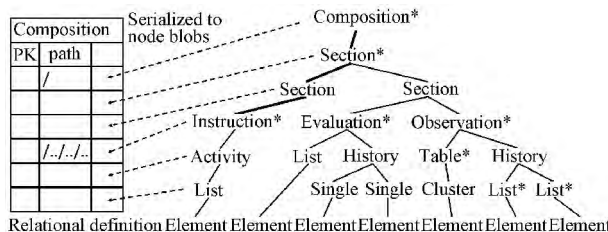


图 1 属性 + 路径数据持久化方法

Fig. 1 Node + Path data persistence method

本研究提出一种基于 openEHR 的原型关系映射 (archetype relational mapping, ARM) 方法, 将 openEHR 原型映射为关系数据库表, 实现原型数据的高效存取、组织和管理。与属性 + 路径和参考模型对象关系映射方法相比, 该方法具有如下特点: 直接在原型和关系数据表之间建立映射关系, 通过一组映射规则来进行数据存储和访问操作。这样, 不同原型实例的数据分别存储在对应原型映射的数据表中, 充分利用了关系数据库的特点, 提高了数据的存储和访问性能。当医疗信息需求发生变化时, 依据原型的版本管理机制对原型进行新增和修订, 形成新版的原型, 并通过原型关系映射方法自动映射到新的关系数据库表, 使数据存储能够适应信息需求的变化, 同时能够保留旧有原型对应的历史数据。基于 openEHR 原型关系映射方法设计实现医疗信息系统, 可以遵循 openEHR 规范, 采用原型结构化组织表达医疗信息, 采用两层建模技术来适应医疗信息需求的变化, 利用 openEHR 原型版本管理机制来保持医疗信息系统升级时数据组织表达的连续性和提供历史数据的持续访问服务。

1 方法

1.1 原型关系映射

openEHR 原型遵循面向对象思想制定, 因此原型关系映射方法可以基于对象关系映射方法实现。Hibernate 是一个成熟的开源对象关系映射项目^[8], 实现了 Java 对象到关系数据库的映射。

原型关系映射方法基于 Hibernate 对象关系映射方法,针对原型的特点进行了修改,解决了原型与关系数据库表之间映射的“阻抗匹配”问题。映射过程如下:首先读取原型和原型关系映射配置,在原型关系映射配置中查找需要映射为关系数据库表字段的原型属性,然后在原型中获取属性的数据类型等信息,最后生成创建关系数据库表的 SQL 语句。主要映射规则包括:

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。

1.2 原型版本控制

openEHR 规范中描述了原型的版本控制机制,以适应医疗信息需求的复杂性和动态性。在 openEHR 规范中,版本号是原型 ID 的重要组成部分。当一个原型正式发布以后,对这个原型的任何修改都要在新版本原型中实现。这样可以保证原型在使用过程中当前版本的稳定性和历史版本的兼容性。一个原型概念可以对应到一系列相同原型的不同版本,保证了使用原型表达数据的连续性。这种做法的一个潜在问题是:随着医疗信息需求的发展变化,对原型的修改会累计产生大量不同版本的原型同时存在。因此,在制定原型的过程中,必须充分考虑各种医疗信息需求,使原型的应用范围尽量广泛;并且,随着旧版本原型应用的逐渐减少而不断淘汰,原型的数量会保持在稳定的

程度。

在原型关系映射方法中,基于关系数据库的限制和约束,对于同一原型的不同版本的映射过程主要分为可兼容升级和不兼容升级两种。可兼容升级主要包括原型属性新增、原型属性废弃、原型属性名变更等,不兼容升级主要包括原型属性类型变化等。原型版本的可兼容升级仍可映射到原有关系数据库表,原型版本的不兼容升级需要映射到新的关系数据库表。主要映射规则包括:

1) 每个新增的原型属性映射为一个新增的关系数据库表字段。

2) 每个废弃的原型属性对应的关系数据库表字段仍然保留。

3) 每个变更名称的原型属性映射为原有关系数据库表字段。

4) 对于原型版本的不兼容升级,按照原型关系映射方法映射到新的关系数据库表。

1.3 原型数据查询语言

openEHR 提出了一种原型数据查询语言 (archetype query language, AQL),用于原型数据查询,如表 1 所示。AQL 的一个主要特性是与系统平台无关,独立于具体应用、编程语言、系统环境和数据存储。但是,openEHR 只定义了数据查询操作的 SELECT 语句,没有数据插入、更新和删除等操作,因此需要直接对原型对象进行相应操作。这样,已经从 AQL 的系统平台无关的层面下降到了与具体平台实现相关的层面,与 AQL 的目标不符,并且很难满足实际医疗环境中存在的多种系统平台的数据访问需求。

表 1 openEHR 原型数据查询语言
Tab. 1 AQL syntax proposed by openEHR

子句	关键词	参数
SELECT	SELECT	Attribute identify path in archetype
	FROM	Archetype name
	WHERE	Attribute identify path in archetype operator (> , > = , = , < , < = , ! =) condition value
	ORDER BY	Attribute identify path in archetype

本研究的目标是提供基于原型的、完整的、与系统平台无关的数据操作,因此必须对 AQL 进行扩充,增加相应的数据插入、更新和删除语句。本研究基于 Hibernate Query Language (HQL),对原型数据查询语言进行了扩展,增加了 INSERT、UPDATE 和 DELETE 等语句,使其具有完整的数据操作功

能 能够 满足 各种 数据 访问 需求 ,如 表 2 所示。openEHR 使用 了 与 平台 无关 的 原型 数据 定义 语言 (data archetype define language , dADL) 来 表达 原型 实例 数据 ,如 SELECT 操作 的 结果 和 INSERT 操作 的 参数。

表 2 基于原型关系映射扩展的原型数据查询语言

Tab.2 AQL syntax extended based on ARM

子句	关键词	参数
INSERT	INSERT	Archetype instances in the format of dADL
UPDATE	UPDATE	Archetype name
	SET	Attribute identify path in archetype operator (=) condition value
DELETE	WHERE	Attribute identify path in archetype operator (> , > = , = , < , < = , ! =) condition value
	DELETE FROM	Archetype name
	WHERE	Attribute identify path in archetype operator (> , > = , = , < , < = , ! =) condition value

在 AQL 中 ,对 一个 原型 的 历史 数据 访问 提供 了 统一 的 查询 形式。对 一个 原型 进行 查询 时 ,如果 不 指定 原型 的 版本 ,将 返回 这个 原型 对应 的 所有 版本 的 数据 ,可以 大大 降低 数据 查询 时 的 复杂性。

2 结果

2.1 数据持久化性能比较

医疗 信息 系统 能够 在 实际 医疗 环境 中 使用 ,必须 要 达到 足够 的 性能 才能 支撑 临床 业务。笔者 对 原型 关系 映射、属性 + 路径 和 参考 模型 对象 关系 映射 等 持久 化 方法 进行 了 性能 测试。测试 选用 在 实际 医疗 环境 中 应用 的 病人 检验 检查 业务 流程 ,包括 医生 下达 检验 检查 申请、病人 执行 检验 检查、医生 根据 检验 检查 数据 形成 报告 等 过程 ,都 具有 良好 设计 的 数据 存储 概念 模型。3 种 持久 化 方法 的 具体 实现 都是 基于 同样 的 数据 存储 概念 模型。根据 临床 信息 需求 制定 的 原型 如 表 3 所示。

测试 数据 集 共有 5 000 条 病人 记录、2 593 条 病人 检查 申请 记录、1 935 条 病人 检查 项目 记录、128 141 条 病人 检查 图像 记录、1 231 条 病人 检查 报告 记录、1 786 条 病人 检验 申请 记录、13 598 条 病人 检验 结果 记录 ,分别 采用 原型 关系 映射、属性 + 路径、参考 模型 对象 关系 映射 3 种 持久 化 方法 存储。属性 +

表 3 检查检验原型

Tab.3 Lab test and examination archetypes

原型
openEHR-EHR-OBSERVATION. exam_data. v1
openEHR-EHR-OBSERVATION. exam_item. v1
openEHR-EHR-OBSERVATION. exam_master. v1
openEHR-EHR-OBSERVATION. exam_report. v1
openEHR-EHR-OBSERVATION. exam_request. v1
openEHR-EHR-OBSERVATION. lab_test_data. v1
openEHR-EHR-OBSERVATION. lab_test_master. v1
openEHR-EHR-OBSERVATION. lab_test_request. v1

路径 数据 存储 模型 采用 一张 表 存储 所有 的 属性 值 和 路径 对 ,采用 最 细 粒度 的 存储 格式 ,即以 原型 叶子 节点 中 的 基本 数据 类型 属性 作为 node 部分 ,以 这个 属性 的 路径 作为 Path 部分 ,以及 所属 原型 对象 的 ID 和 原型 名 作为 一条 记录 ,包含 6 415 566 条 记录。参考 模型 对象 关系 映射 数据 存储 模型 共有 156 张 表 ,涵盖 了 整个 openEHR 参考 模型。其中 ,DataValue 是 所有 基本 数据 类型 的 基类 ,包含 6 456 406 条 记录。而 大部分 原型 属性 的 类型 为 字符串 DvText ,包含 5 202 899 条 记录。在 原型 关系 映射 数据 存储 模型 中 ,包含 由 原型 映射 生成 的 8 个 数据 表 ,其中 记录 数量 与 测试 数据 集 规模 和 分布 相同。由于 在 属性 + 路径、参考 模型 对象 关系 映射 两种 数据 存储 模型 中 ,大量 数据 集中 于 少量 数据 表 ,并且 存在 许多 相同 的 记录 值 ,添加 索引 不能 有效 提高 数据 访问 性能 ,并且 会 带来 写入 性能 的 急剧 下降 ,因此 3 种 持久 化 方法 的 关系 数据库 中 都 未 添加 索引。测试 电脑 配置 ,CPU 为 Intel Core i5-3340M ,内存 16 GB ,硬盘 128 GB SSD ,操作系统 为 Windows 8.1 Enterprise 64 位 ,数据库 为 SQL Server 2012 Enterprise 64 位。

具体 选择 查询 2014 年 1 月 1 日 的 病人 就诊 列表 信息 ,以 病历 号为 X 的 病人 所有 检查 和 检验 信息 作为 测试 用例 ,分别 对 病人 就诊 列表、检验 申请 记录、检验 结果 记录、检查 申请 记录、检查 图像 数据、检查 报告 数据 等 使用 单个 条件 进行 查询 ,并 得到 查询 结果 的 原型 对象 实例。病人 就诊 列表 共 包含 165 条 记录 ,X 病人 共有 21 条 检验 申请 记录、199 条 检验 结果 记录、5 条 检查 申请 记录、1 593 条 检查 图像 记录、3 条 检查 报告 记录 ,所有 查询 均 进行 3 次 取 平均 值。原型 关系 映射 数据 查询 时间 见表 4 ,属性 + 路径 数据 查询 时间 见表 5 ,参考 模型 对象 关系 映射 数据 查询 时间 见表 6 ,平均 数据 查询 时间 见表 7。

表 4 原型关系映射数据查询时间(ms)

Tab. 4 ARM data query time (ms)

查询	T1	T2	T3
病人就诊	1 072	1 112	1 083
检验申请	258	263	269
检验结果	1 405	1 391	1 367
检查申请	58	61	60
检查图像	8 544	8 461	8 540
检查报告	34	30	32

表 5 属性 + 路径数据查询时间(ms)

Tab. 5 Node + Path data query time(ms)

查询	T1	T2	T3
病人就诊	4 304	4 121	4 139
检验申请	2 592	2 642	2 645
检验结果	4 616	4 674	4 673
检查申请	2 283	2 298	2 295
检查图像	17 770	17 672	17 613
检查报告	2 220	2 247	2 244

表 6 参考模型对象关系映射数据查询时间(ms)

Tab. 6 openEHR RM ORM data query time(ms)

查询	T1	T2	T3
病人就诊	12 430	12 575	12 077
检验申请	1 523	1 525	1 585
检验结果	15 906	15 958	15 800
检查申请	418	429	424
检查图像	147 551	143 357	148 033
检查报告	252	271	260

表 7 平均数据查询时间(ms)

Tab. 7 Average data query time (ms)

查询	原型关系映射	属性 + 路径	参考模型对象关系映射
病人就诊	1 089	4 188	12 361
检验申请	263	2 626	1 544
检验结果	1 388	4 654	15 888
检查申请	60	2 292	424
检查图像	8 515	17 685	146 313
检查报告	32	2 237	261

以上结果表明,在 3 种持久化方法中,原型关系映射方法的性能最好。原型关系映射方法通过在原型与关系数据表之间建立对应关系,将原型数据分散存储到不同的数据表中,有效利用了关系数据库的优点,避免了其他两种持久化方法中数据集中在少量数据表中造成的不必要的数据全表扫描操作。在常用的数据库优化技术中,分表是一种常用技术,通过将单个表中的数据按照一定逻辑分散到多个表中来提高查询效率,因此将数据分散到多个表中对效率的提升具有显著作用。在属性 + 路径

数据存储模型中,数据集中在一个表中,所有数据访问都需要进行全表扫描,因此在查询少量数据的情况下性能也很差。在参考模型对象关系映射数据存储模型中,由于参考模型包含大量的类,类之间具有深层的继承关系和复杂的组合关系,对于一条记录的查询需要生成 21 个 SELECT 语句,这些语句共包含 1 250 个 JOIN 操作,虽然对象关系映射框架会对生成的 SQL 语句进行一定优化,但是大量的 JOIN 操作需要消耗大量的计算能力,导致随着数据查询量的增加性能急剧降低。在实际医疗环境中,查询通常包括多个条件。在原型关系映射方法中,如果多个条件属于同一个原型,则对查询性能没有太大影响,如果多个条件属于原型中的集合类型,查询过程中会产生多个数据表的 JOIN 操作,查询性能会受到 JOIN 表数量和表中记录数量的影响。在属性 + 路径数据存储模型中,对于多个条件的查询有两种方式,一种是将一个复杂查询的多个条件拆解后反复进行单条件查询,一种是根据查询条件的数量将属性 + 路径数据表进行多次 JOIN 操作,在这个集合中进行多个条件的查询。由于属性 + 路径数据表包含了所有原型对象的数据,这两种方法与原型关系映射方法相比都不具有性能优势。在参考模型对象关系映射数据存储模型中,由于参考模型中没有提供原型组合关联的语法,因此不能实现原型之间的关联查询;如果多个条件属于同一个原型,则在查询过程中也会产生多个数据表的 JOIN 操作,查询性能会受到 JOIN 表数量和表中记录数量的影响。查询结果中的原型可能会包含集合类型或与关联原型。在原型关系映射方法中,当需要完整还原一个包含集合类型或与关联原型的原型时,需要通过 JOIN 操作实现,查询性能会受到 JOIN 表数量和表中记录数量的影响。在属性 + 路径数据存储模型中,需要先查询主原型,再查询集合类型或关联原型,分步查询的性能比原型关系映射方法使用 JOIN 操作的性能要低。在参考模型对象关系映射数据存储模型中,不支持原型之间的关联,不能实现关联原型的查询。

2.2 数据服务平台

笔者基于原型关系映射方法、原型数据查询语言、原型数据定义语言等技术,利用 Web Service 开发了数据服务平台,提供数据查询 Select、数据插入 Insert、数据更新 Update 和数据删除 Delete 等 4 种数据操作接口,使用原型数据查询语言作为数据操作语言,利用原型数据定义语言作为与平台无关的数

据表达形式。医护人员只要根据信息需求制定出相应的原型,就可以在数据服务平台中存储和访问数据。目前,示范应用的临床业务数据包括病人基本信息、检查检验信息以及各类专科疾病信息。下面以老年痴呆症(Alzheimer's disease, AD)为例,介绍数据服务平台的特点。

老年痴呆症是典型的专科疾病,具有特有的医疗信息需求。随着医学研究的不断深入,新的诊断方法不断涌现^[9],医疗信息需求也随之不断发展变化。原有老年痴呆症采用纸质记录,主要为一系列智力测试问卷,包括日常生活活动(activity of daily livings, ADL)、临床痴呆评定量表(clinical dementia rating, CDR)、老年抑郁量表(geriatric depression scale, GDS)、简易智能状态检查(mini mental state exam, MMSE)、蒙特利尔认知评估量表(Montreal cognitive assessment, MoCA)等。笔者针对每个问卷,制定了相应的原型。

数据服务平台利用问卷原型生成信息录入界面,采用原型数据定义语言表达医护人员录入的问卷信息,利用原型关系映射方法将信息存储到关系数据库中,并通过原型数据访问语言提供问卷数据访问。通过对原型的新增和修订,可以灵活适应医疗信息需求的变化。

3 讨论和结论

笔者基于 openEHR 规范设计了原型关系映射方法,并与属性+路径和参考模型对象关系映射等持久化方法进行了测试和对比。结果表明,本研究提出的原型关系映射方法的性能比属性+路径和参考模型对象关系映射方法的性能更好。该方法扩展了原型数据查询语言,能够满足各种医疗数据访问需求。对数据服务平台进行了验证,表明基于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。

由于医疗信息的复杂性,典型的医疗信息系统通常包含上百个原型,这些原型以及原型之间的复杂关系会映射生成上百个关系数据库表。原型的设计对于数据访问效率具有较大影响,尤其对于原型中的集合、嵌套等表达形式,在数据访问过程中会涉及多张表的联合操作,这与传统关系库设计过程中遇到的性能优化问题相同,原型关系映射方法

在生成关系数据库的过程中并未考虑像传统数据库设计过程中常用的反范式等策略。原型作为表达医疗信息领域概念的载体,需要医疗专家和 IT 专家的密切配合,遵循低耦合、高内聚的原则制定出良好设计的原型,在保证充分清晰表达医疗概念的同时,采用更加简单的表达形式。精心设计的原型是宝贵的医疗信息资源,需要进行统一管理和维护,使其逐步积累形成原型仓库,成为构建医疗信息系统和进行医疗信息交换的重要基础。

原型关系映射方法从表达医疗领域信息概念的原型出发,利用原型生成基于关系数据库的数据存储和驱动数据访问,将医疗信息需求与医疗信息系统分离,使医疗信息系统能够利用原型实现对信息需求快速变化的适应,满足各类不断发展变化的专科疾病特有的信息管理需求。与基于传统关系数据库方法的医嘱管理系统等性能要求较高和信息需求较为固定的医疗信息系统相比,原型关系映射方法牺牲了根据具体查询进行人工性能优化的需求,在性能上做了一定的妥协。

参考文献

- [1] Haux R. Health information systems -past, present, future [J]. *International Journal of Medical Informatics*, 2006, **75**(3-4): 268-281.
- [2] Mandl KD, Kohane IS. Escaping the EHR Trap — The future of health IT [J]. *N Engl J Med*, 2012, **366**(24): 2240-2242.
- [3] Kalra D, Beale T, Heard S. The openEHR Foundation [J]. *Stud Health Technol Inform*, 2005, **115**: 153-173.
- [4] 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.
- [5] 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.
- [6] Späth 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.
- [7] Freire SM. 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] Hibernate. Hibernate [EB/OL]. <http://www.hibernate.org/>, 2011-12-15/2014-06-29.
- [9] Thies W, Bleiler L. 2013 Alzheimer's disease facts and figures [J]. *Alzheimers Dement*, 2013, **9**(2): 208-245.