



(12)发明专利

(10)授权公告号 CN 105045933 B

(45)授权公告日 2018.04.24

(21)申请号 201510565705.2

(22)申请日 2015.09.08

(65)同一申请的已公布的文献号

申请公布号 CN 105045933 A

(43)申请公布日 2015.11.11

(73)专利权人 中国人民解放军海军工程大学
地址 430033 湖北省武汉市解放大道717号

(72)发明人 马良荔 孙煜飞 李永杰 苏凯
覃基伟 许国鹏

(74)专利代理机构 武汉开元知识产权代理有限公司 42104

代理人 黄行军 李满

(51)Int.Cl.

G06F 17/30(2006.01)

(56)对比文件

CN 104036048 A, 2014.09.10,

CN 101149748 A, 2008.03.26,

US 7877421 B2, 2011.01.25,

张旭辉.RDF_RDFS到关系数据库模式映射方法的研究.《中国优秀硕士学位论文全文数据库信息科技辑》.2012,(第6期),I138-1590.

周扬.基于关系数据库的本体映射方法研究.《中国优秀博硕士学位论文全文数据库(硕士)信息科技辑》.2006,(第10期),I138-334.

郭朝敏 等.一种关系数据库到本体的自动构建方法.《计算机工程与应用》.2012,第48卷(第7期),第115-120,248页.

审查员 刘亭

权利要求书4页 说明书9页 附图2页

(54)发明名称

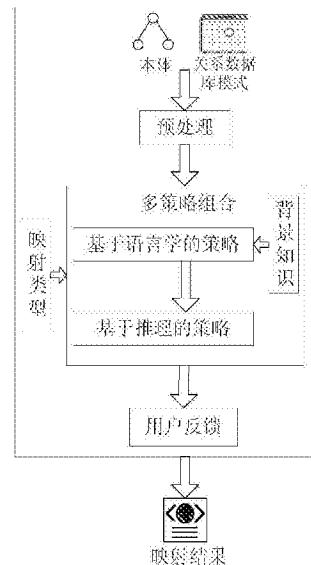
船舶装备维修保障信息关系数据库模式与本体间映射方法

(57)摘要

本发明公开了一种船舶装备维修保障信息关系数据库模式与本体间映射方法,包括如下步骤:步骤1:对船舶装备维修保障信息本体的元素和船舶装备维修保障信息关系数据库模式的元素进行分类,建立船舶装备维修保障信息关系数据库模式和船舶装备维修保障信息本体之间的映射类型;步骤2:利用多策略组合的方法得到待处理船舶装备维修保障信息关系数据库模式与待处理船舶装备维修保障信息本体之间的映射关系。本发明能够有效地集成现有的船舶装备维修保障信息,便于维修保障人员对其实施有效管理,提供决策支持和远程维修保障支援能力,推进维修保障信息的高效利用和保障资源的优化,使装备发挥其最大效能,因而具有重要理论价值和实际意义。

B

CN 105045933



CN

1. 一种船舶装备维修保障信息关系数据库模式与本体间映射方法，其特征在于，它包括如下步骤：

步骤1：对船舶装备维修保障信息本体的元素和船舶装备维修保障信息关系数据库模式的元素进行分类，将上述船舶装备维修保障信息关系数据库模式的元素中的关系R分为实体类型关系ER和关系类型关系RR，将船舶装备维修保障信息关系数据库模式中的属性A按数据库逆向工程方式分为外键属性FKA和非外键属性NFKA，将船舶装备维修保障信息本体的元素按船舶装备维修保障信息本体自带的说明规范，分为船舶装备维修保障信息本体元素的类C和属性P，将属性P进一步分为船舶装备维修保障信息本体元素的对象属性P_o和船舶装备维修保障信息本体元素的数据类型属性P_d，建立船舶装备维修保障信息关系数据库模式和船舶装备维修保障信息本体之间的如下映射类型：

实体类型关系ER至船舶装备维修保障信息本体元素的类C的映射类型；

关系类型关系RR至船舶装备维修保障信息本体元素的对象属性P_o的映射类型；

外键属性FKA至船舶装备维修保障信息本体元素的对象属性P_o的映射类型；

非外键属性NFKA至船舶装备维修保障信息本体元素的对象属性P_o与船舶装备维修保障信息本体元素的数据类型属性P_d并集的映射类型；

步骤2：根据如下步骤201～步骤203得到待处理船舶装备维修保障信息关系数据库模式与待处理船舶装备维修保障信息本体之间的映射关系：

步骤201：将待处理船舶装备维修保障信息本体的元素的命名方式统一为英文描述，然后判断船舶装备维修保障信息关系数据库模式的元素的命名方式为以下两种命名方式的哪一种：

第一种船舶装备维修保障信息关系数据库模式的元素的命名方式为：使用英文单词或缩写进行描述；

第二种船舶装备维修保障信息关系数据库模式的元素的命名方式为：直接使用中文进行描述；

步骤202：对于第二种命名方式，首先将对应的船舶装备维修保障信息关系数据库模式的元素转换成英文描述，此时即将第二种命名方式转换成了第一种命名方式，然后为待处理船舶装备维修保障信息本体的每个元素和待处理船舶装备维修保障信息关系数据库模式的每个元素分别构建对应的虚拟文档，每个虚拟文档包括上述待处理船舶装备维修保障信息本体的每个元素或船舶装备维修保障信息关系数据库模式的每个元素对应的英文描述，还包括与上述每个元素结构上相邻的元素对应的英文描述，根据步骤1的分类方式为不同类型船舶装备维修保障信息本体的元素和船舶装备维修保障信息关系数据库模式的元素构建不同的虚拟文档；

为待处理船舶装备维修保障信息关系数据库模式的元素中的关系，根据以下方式构建虚拟文档：

对于待处理船舶装备维修保障信息关系数据库模式中的关系R，若为实体类型关系ER，则该关系R的虚拟文档只包括自身的英文描述；若为关系类型关系RR，则该关系R的虚拟文档不仅包括其自身的英文描述，还包括关系类型关系RR所引用关系的英文描述；

用vdoc()函数表示虚拟文档，用desc()函数表示待处理船舶装备维修保障信息本体的每个元素或船舶装备维修保障信息关系数据库模式的每个元素的英文描述，具体形式化

如下：

$$vdoc(R) = \begin{cases} desc(R) & R \in ER \\ desc(R) + desc(rel(A'_R)) & R \in RR, A'_R \in ref(A_R), A_R \in pk(R) \end{cases} \quad (1)$$

其中, $A_R \in pk(R)$ 表示 A_R 属于关系 R 的主键, $A'_R \in ref(A_R)$ 表示关系 R 的主键引用的属性, $rel(A'_R)$ 表示属性 A'_R 所属的关系, $desc(rel(A'_R))$ 表示属性 A'_R 所属的关系的英文描述, $vdoc(R)$ 表示关系 R 的虚拟文档, $desc(R)$ 表示关系 R 的英文描述;

对于待处理船舶装备维修保障信息关系数据库模式中的属性 A , 则其虚拟文档除了自身的英文描述外, 还包括该属性 A 所属的关系的英文描述, 如果它是外键属性 FKA, 则还需进一步考虑该属性 A 引用的属性所属关系的描述; 如果该属性 A 是非外键属性 NFKA, 则需补充其数据类型, 具体如下:

$$vdoc(A) = \begin{cases} desc(A) + desc(rel(A)) + desc(rel(A')) & A' \in ref(A), A \in FKA \\ desc(A) + desc(rel(A)) + desc(type(A)) & A \in NFKA \end{cases} \quad (2)$$

其中, $type(A)$ 表示属性 A 的数据类型, $desc(type(A))$ 表示属性 A 数据类型的英文描述, $desc(rel(A))$ 表示属性 A 所属的关系的英文描述, $desc(A)$ 表示属性 A 的英文描述, $desc(rel(A'))$ 表示属性 A' 所属关系的英文描述;

对于船舶装备维修保障信息本体元素的类 C , 它的虚拟文档就是其自身的英文描述, 即:

$$vdoc(C) = desc(C) \quad (3)$$

对于待处理船舶装备维修保障信息本体中的属性 P , 则其虚拟文档除了自身的自然语言描述外, 还包括它的定义域 $d(P)$ 和值域 $r(P)$ 的自然语言描述, 对于船舶装备维修保障信息本体元素的数据类型属性 P_d , 其值域为数据类型, 具体如下:

$$vdoc(P) = desc(P) + desc(d(P)) + desc(r(P)) \quad (4)$$

其中, 待处理船舶装备维修保障信息本体中的属性 P = 船舶装备维修保障信息本体元素的数据类型属性 P_d + 船舶装备维修保障信息本体元素的对象属性 P_o , $vdoc(P)$ 表示待处理船舶装备维修保障信息本体中属性 P 的虚拟文档, $desc(P)$ 表示待处理船舶装备维修保障信息本体中属性 P 的英文描述, $desc(d(P))$ 表示定义域 $d(P)$ 的英文描述, $desc(r(P))$ 表示值域 $r(P)$ 的英文描述;

这样, 就为每个船舶装备维修保障信息关系数据库模式和本体的元素构建了虚拟文档, 每个虚拟文档都看成向量空间模型中的一个向量, 对于每个虚拟文档, 其中任一单词 i 的权重用如下 TF-IDF 算法进行计算:

$$tf_i = \frac{n_i}{\sum_k n_k} \quad (5)$$

$$idf_i = \log \frac{N}{n+1} \quad (6)$$

$$w_i = tf_i \cdot idf_i \quad (7)$$

其中 tf_i 为单词 i 的词频, n_i 为单词 i 在该虚拟文档内出现的次数, $\sum_k n_k$ 为该虚拟文档的

总词数; idf_i 为逆文档频率, N 为虚拟文档总数, n 为含有单词 i 的虚拟文档数, w_i 表示单词 i 的权重, 这样虚拟文档就转换成带权值的单词集合即向量, 对于任意两个带权值的单词集合, 即向量 e_i 和 e_j 之间的相似度 $sim(e_i, e_j)$ 用如下余弦相似度公式来衡量:

$$sim(e_i, e_j) = \frac{\sum_{k=1}^D w_{ik} w_{jk}}{\sqrt{\sum_{k=1}^D w_{ik}^2} \sqrt{\sum_{k=1}^D w_{jk}^2}} \quad (8)$$

其中, w_{ik} 表示向量 e_i 中单词 k 的权值, w_{jk} 表示向量 e_j 中单词 k 的权值, D 为向量的维度, 即所有单词的总数;

步骤203: 按照步骤1建立的映射类型, 和步骤201、步骤202中的方法, 计算船舶装备维修保障信息关系数据库模式中元素 e_s 与本体中元素 e_t 之间的相似度 $sim(e_s, e_t)$, 即船舶装备维修保障信息关系数据库模式的元素对应的向量之间的相似度, 得到所有满足映射类型要求的元素间相似度后, 按照相似度从大到小排序, 设定阈值 θ ($\theta \in (0, 1)$), 过滤掉所有相似度小于 θ 的元素对, 对于相似度大于或等于 θ 的元素对, 建立这两个元素之间的映射关系, 即得到船舶装备维修保障信息关系数据库模式与本体间的映射。

2. 根据权利要求1所述的船舶装备维修保障信息关系数据库模式与本体间映射方法, 其特征在于, 所述步骤203后还包括步骤3:

为了发现步骤203中得到的船舶装备维修保障信息关系数据库模式与本体间的映射是否存在错误, 并将存在的错误排除, 设计了基于推理的策略, 主要包括如下两条推理规则:

如果存在待处理船舶装备维修保障信息关系数据库模式中的属性 A 与待处理船舶装备维修保障信息本体中的属性 P 之间的映射 m , 但不存在所述属性 A 所属的船舶装备维修保障信息关系数据库模式的元素中的关系 R 与所述属性 P 定义域所属的类 C 之间的映射, 则映射 m 为错误映射;

如果存在待处理船舶装备维修保障信息关系数据库模式中的属性 A 与船舶装备维修保障信息本体元素的数据类型属性 P_d 之间的映射 m_1 , 但是所述属性 A 的数据类型与所述数据类型属性 P_d 的值域所属的数据类型为预设的不兼容数据类型, 则 m_1 为错误映射;

在船舶装备维修保障信息关系数据库模式与本体间的映射结果中删除上述错误映射 m 和错误映射 m_1 。

3. 根据权利要求2所述的船舶装备维修保障信息关系数据库模式与本体间映射方法, 其特征在于, 所述步骤3后还包括步骤4:

为了进一步提高上述得到的删除错误映射 m 和错误映射 m_1 后的船舶装备维修保障信息关系数据库模式与本体间映射的准确率和查全率, 设计了如下系统提问、用户回答的交互方式来排除更多错误映射, 从而提高映射结果的质量, 提问规则为:

规则一、如果存在船舶装备维修保障信息关系数据库模式和本体间映射 (e_s, e_t) 和映射 (e_s', e_t') , 且其中 $e_s = e_s'$ 或 $e_t = e_t'$, e_s, e_s' 为船舶装备维修保障信息关系数据库模式中的元素, e_t, e_t' 为船舶装备维修保障信息本体的元素, 则映射 (e_s, e_t) 和映射 (e_s', e_t') 中有一个映射为错误映射, 选择相似度差值最小的映射, 即

$$Q(e_s, e_t) = \min \{ |sim(e_s, e_t) - sim(e_s', e_t')| \}; e_s = e_s' \text{ 或 } e_t = e_t'$$

规则二、选择相似度 $sim(e_s, e_t)$ 与阈值 θ 相差最小的映射，即

$$Q(e_s, e_t) = \min \{ |sim(e_s, e_t) - \theta| \}$$

该方式通过不断迭代，每次迭代选择满足上述两条规则的映射，用户对映射正确与否做出判断，使得系统能够在最少问答次数的情况下，获得最大的映射质量增加。

船舶装备维修保障信息关系数据库模式与本体间映射方法

技术领域

[0001] 本发明涉及船舶装备维修保障信息异构数据库的数据集成技术领域,具体涉及一种船舶装备维修保障信息关系数据库模式与本体间映射方法。

技术背景

[0002] 随着信息技术的迅速发展,有关船舶装备维修保障的信息系统越来越多,不同组织、部门由于业务与功能需求的不同,各自都开发了相应的信息管理系统,这些系统之间数据环境各异,数据描述千差万别,形成了一个个的“数据孤岛”。“数据孤岛”造成了信息系统之间存在着大量的冗余数据、垃圾数据,无法保证数据的一致性,信息的利用率低。开发新的集成领域内所有信息的管理系统耗时耗力,因此,如何有效地重用这些信息,实现数据的共享与互操作,为维修保障人员提供更有效的决策支持,是当前信息化建设的一个迫切课题。

[0003] 数据集成一直以来都是计算机领域、尤其是数据库领域的研究重点和难点。数据集成是指将互相关联的分布式异构数据源集成在一起,使用户能以透明的方式对这些数据源进行统一访问。传统的数据集成方式包括:多数据库系统、联邦数据库系统、数据仓库、基于包装器和协调器的数据集成、基于agent的数据集成、基于web服务的数据集成、P2P数据集成以及数据空间等。在以上代表性的研究工作中,注重的都是数据结构的集成,几乎没有涉及到语义的集成。

[0004] 语义Web提供了一个通用框架,使得数据的共享和重用可以跨越应用系统、企业和社区的边界。本体是语义Web的核心基础元素,它作为一种领域知识概念化和模型化的方法,可以用来描述计算机处理数据的语义信息,被认为是解决语义异构的关键途径。近年来,有关本体的表示(RDFS/OWL)、查询语言(SPARQL)等各种技术日趋成熟,特别是链接开放数据项目(linked open data)的大力推动,标志着以本体为核心语义Web基础技术已经奠定。

[0005] 目前,船舶维修保障信息绝大部分仍以关系数据库的方式进行存储,利用本体对这些数据进行语义集成,首先就必须发现关系数据库模式与本体之间的映射关系,然后才能实施查询处理或数据转换,最终实现语义数据集成。因此,建立船舶装备维修保障信息关系数据库模式与本体之间的映射关系,是建立基于本体的语义数据集成系统的关键。

[0006] 但是,关系数据库模式和本体这两种不同的模型在建模思想、应用场景上有着很大的差别,关系数据库模式是对具体数据的抽象描述,针对特定的应用程序而设计,规模较小,语义表达能力较弱,由程序的开发者独立创建;而本体则试图建立领域内的共享概念模型,独立于具体应用,规模较大,语义表达能力较强,由领域专家共同创建。因此,关系数据库模式与本体之间映射比一般的同种异构模型映射(例如模式匹配、本体映射)显得更加困难。

[0007] 目前,船舶装备维修保障信息关系数据库模式与本体之间的映射研究大部分仍采用人工定义的方式,部分项目的研究中还开发了相应的图形用户界面,支持交互式的关系

数据库模式与本体间映射的构建,例如DartGrid、MapOnto等。人工构建映射的方式耗时耗力且容易出错,因此,也有部分工作针对自动化映射建立展开了研究,例如申请号为201410314898.X的中国发明发明“一种本体与关系数据库模式之间的映射方法”。该专利记载的完全自动化的方法不一定有效,并且当关系数据库模式与本体规模增大时,基于结构相似度传播的方法时间复杂度过高,无法接受。

发明内容

[0008] 本发明的目的是针对上述技术问题,提供一种船舶装备维修保障信息关系数据库模式与本体间映射方法,该方法能够有效地集成现有的船舶装备维修保障信息,便于维修保障人员提高船舶装备维修保障效率。

[0009] 为实现此目的,本发明所设计的船舶装备维修保障信息关系数据库模式与本体间映射方法,其特征在于,它包括如下步骤:

[0010] 步骤1:对船舶装备维修保障信息本体的元素和船舶装备维修保障信息关系数据库模式的元素进行分类,将上述船舶装备维修保障信息关系数据库模式的元素中的关系R分为实体类型关系ER和关系类型关系RR,将船舶装备维修保障信息关系数据库模式中的属性A按数据库逆向工程方式分为外键属性FKA和非外键属性NFKA,将船舶装备维修保障信息本体的元素按船舶装备维修保障信息本体自带的说明规范,分为船舶装备维修保障信息本体元素的类C和属性P,将属性P进一步分为船舶装备维修保障信息本体元素的对象属性P_o和船舶装备维修保障信息本体元素的数据类型属性P_d,建立船舶装备维修保障信息关系数据库模式和船舶装备维修保障信息本体之间的如下映射类型:

[0011] 实体类型关系ER至船舶装备维修保障信息本体元素的类C的映射类型;

[0012] 关系类型关系RR至船舶装备维修保障信息本体元素的对象属性P_o的映射类型;

[0013] 外键属性FKA至船舶装备维修保障信息本体元素的对象属性P_o的映射类型;

[0014] 非外键属性NFKA至船舶装备维修保障信息本体元素的对象属性P_o与船舶装备维修保障信息本体元素的数据类型属性P_d并集的映射类型;

[0015] 步骤2:根据如下步骤201~步骤203得到待处理船舶装备维修保障信息关系数据库模式与待处理船舶装备维修保障信息本体之间的映射关系:

[0016] 步骤201:将待处理船舶装备维修保障信息本体的元素的命名方式统一为英文描述,然后判断船舶装备维修保障信息关系数据库模式的元素的命名方式为以下两种命名方式的哪一种:

[0017] 第一种船舶装备维修保障信息关系数据库模式的元素的命名方式为:使用英文单词或缩写进行描述;

[0018] 第二种船舶装备维修保障信息关系数据库模式的元素的命名方式为:直接使用中文进行描述;

[0019] 步骤202:对于第二种命名方式,首先将对应的船舶装备维修保障信息关系数据库模式的元素转换成英文描述,此时即将第二种命名方式转换成了第一种命名方式,然后为待处理船舶装备维修保障信息本体的每个元素和待处理船舶装备维修保障信息关系数据库模式的每个元素分别构建对应的虚拟文档,每个虚拟文档包括上述待处理船舶装备维修保障信息本体的每个元素或船舶装备维修保障信息关系数据库模式的每个元素对应的英

文描述,还包括与上述每个元素结构上相邻的元素对应的英文描述,根据步骤1的分类方式为不同类型船舶装备维修保障信息本体的元素和船舶装备维修保障信息关系数据库模式的元素构建不同的虚拟文档;

[0020] 为待处理船舶装备维修保障信息关系数据库模式的元素中的关系,根据以下方式构建虚拟文档:

[0021] 对于待处理船舶装备维修保障信息关系数据库模式中的关系R,若为实体类型关系ER,则该关系R的虚拟文档只包括自身的英文描述;若为关系类型关系RR,则该关系R的虚拟文档不仅包括其自身的英文描述,还包括关系类型关系RR所引用关系的英文描述;

[0022] 用vdoc()函数表示虚拟文档,用desc()函数表示待处理船舶装备维修保障信息本体的每个元素或船舶装备维修保障信息关系数据库模式的每个元素的英文描述,具体形式化如下:

[0023]

$$vdoc(R) = \begin{cases} desc(R) & R \in ER \\ desc(R) + desc(rel(A'_R)) & R \in RR, A'_R \in ref(A_R), A_R \in pk(R) \end{cases} \quad (1)$$

[0024] 其中, $A_R \in pk(R)$ 表示 A_R 属于关系R的主键, $A'_R \in ref(A_R)$ 表示关系R的主键引用的属性, $rel(A'_R)$ 表示属性 A'_R 所属的关系, $desc(rel(A'_R))$ 表示属性 A'_R 所属的关系的英文描述,vdoc(R) 表示关系R的虚拟文档,desc(R) 表示关系R的英文描述;

[0025] 对于待处理船舶装备维修保障信息关系数据库模式中的属性A,则其虚拟文档除了自身的英文描述外,还包括该属性A所属的关系的英文描述,如果它是外键属性FKA,则还需进一步考虑该属性A引用的属性所属关系的描述;如果该属性A是非外键属性NFKA,则需补充其数据类型,具体如下:

[0026]

$$vdoc(A) = \begin{cases} desc(A) + desc(rel(A)) + desc(rel(A')) & A' \in ref(A), A \in FKA \\ desc(A) + desc(rel(A)) + desc(type(A)) & A \in NFKA \end{cases} \quad (2)$$

[0027] 其中,type(A) 表示属性A的数据类型,desc(type(A)) 表示属性A数据类型的英文描述,desc(rel(A)) 表示属性A所属的关系的英文描述,desc(A) 表示属性A的英文描述,desc(rel(A')) 表示属性A'所属关系的英文描述;

[0028] 对于船舶装备维修保障信息本体元素的类C,它的虚拟文档就是其自身的英文描述,即:

[0029] $vdoc(C) = desc(C) \quad (3)$

[0030] 对于待处理船舶装备维修保障信息本体中的属性P,则其虚拟文档除了自身的自然语言描述外,还包括它的定义域d(P) 和值域r(P) 的自然语言描述,对于船舶装备维修保障信息本体元素的数据类型属性P_d,其值域为数据类型,具体如下:

[0031] $vdoc(P) = desc(P) + desc(d(P)) + desc(r(P)) \quad (4)$

[0032] 其中,待处理船舶装备维修保障信息本体中的属性P=船舶装备维修保障信息本体元素的数据类型属性P_d+船舶装备维修保障信息本体元素的对象属性P_o,vdoc(P) 表示待处理船舶装备维修保障信息本体中属性P的虚拟文档,desc(P) 表示待处理船舶装备维修保障信息本体中属性P的英文描述,desc(d(P)) 表示定义域d(P) 的英文描述,desc(r(P)) 表示

值域r(P)的英文描述:

[0033] 这样,就为船舶装备维修保障信息关系数据库模式和本体的每个元素构建了虚拟文档,每个虚拟文档都看成向量空间模型中的一个向量,对于每个虚拟文档,其中任一单词i的权重用如下TF-IDF算法进行计算:

$$[0034] \quad tf_i = \frac{n_i}{\sum_k n_k} \quad (5)$$

$$[0035] \quad idf_i = \log \frac{N}{n+1} \quad (6)$$

$$[0036] \quad w_i = tf_i \cdot idf_i \quad (7)$$

[0037] 其中 tf_i 为单词i的词频, n_i 为单词i在该虚拟文档内出现的次数, $\sum_k n_k$ 为该虚拟文档的总词数; idf_i 为逆文档频率,N为虚拟文档总数,n为含有单词i的虚拟文档数, w_i 表示单词i的权重,这样虚拟文档就转换成带权值的单词集合即向量,对于任意两个带权值的单词集合,即向量 e_i 和 e_j 之间的相似度 $sim(e_i, e_j)$ 用如下余弦相似度公式来衡量:

$$[0038] \quad sim(e_i, e_j) = \frac{\sum_{k=1}^D w_{ik} w_{jk}}{\sqrt{\sum_{k=1}^D w_{ik}^2} \sqrt{\sum_{k=1}^D w_{jk}^2}} \quad (8)$$

[0039] 其中, w_{ik} 表示向量 e_i 中单词k的权值, w_{jk} 表示向量 e_j 中单词k的权值,D为向量的维度,即所有单词的总数;

[0040] 步骤203:按照步骤1建立的映射类型,和步骤201、步骤202中的方法,计算船舶装备维修保障信息关系数据库模式中元素 e_s 与本体中元素 e_t 之间的相似度 $sim(e_s, e_t)$,即船舶装备维修保障信息关系数据库模式的元素对应的向量之间的相似度,得到所有满足映射类型要求的元素间相似度后,按照相似度从大到小排序,设定阈值 θ ($\theta \in (0, 1)$),过滤掉所有相似度小于 θ 的元素对,对于相似度大于或等于 θ 的元素对,建立这两个元素之间的映射关系,即得到船舶装备维修保障信息关系数据库模式与本体间的映射。

[0041] 本发明借鉴信息检索技术,应用多策略组合的关系数据库模式与本体间半自动映射方法,克服了已有关系数据库模式与本体间映射方案的人工定义或复杂度高等缺点,关系数据库模式与本体间映射是建立基于本体的装备维修保障信息语义集成系统的前提,能够有效地集成现有的船舶装备维修保障信息,便于维修保障人员对其实施有效管理,提供决策支持和远程维修保障支援能力,推进维修保障信息的高效利用和保障资源的优化,使装备发挥其最大效能,因而具有重要理论价值和实际意义。

附图说明

[0042] 图1为本发明的关系数据库模式与本体间映射方法流程示意图;

[0043] 图2为本发明的船舶装备维修保障信息故障本体示例;

[0044] 图3为本发明的船舶装备维修保障信息故障关系数据库示例；

[0045] 图4为本发明的关系数据库模式与本体间映射工具界面。

具体实施方式

[0046] 以下结合附图和实施例对本发明作进一步的详细说明：

[0047] 本发明首先，对关系数据库模式及本体的元素类型进行分类，本体元素主要包括类、对象属性和数据类型属性，将关系数据库模式的关系分为实体类型关系和关系类型关系，将关系数据库模式的属性分为外键属性和非外键属性，然后在上述元素分类的基础上，依据启发式规则，建立关系数据库模式元素与本体元素之间的映射类型；

[0048] 然后，综合元素的各种可用信息，利用多种策略发现关系数据库模式与本体元素之间的映射关系。首先对船舶维修保障领域本体的建立进行了介绍，并对船舶维修保障领域关系数据库元素的命名规则进行了分析，将其分为两种情况，分别为英文、拼音和中文描述。然后根据以上分析，对不同的情况进行不同的处理，设计了多种策略发现二者之间的映射关系，包括：基于语言学的策略、基于背景知识的策略和基于推理的策略。基于语言学的策略利用元素的语言描述信息（例如属性的名称、类的注释等），如果关系数据库模式为英文描述，则先将本体元素转换成英文，然后利用向量空间模型和余弦相似度公式计算元素之间的相似度；如果关系数据库元素用中文描述，则先将关系数据库和本体元素转换成英文，然后利用向量空间模型和余弦相似度公式计算元素之间的相似度。基于背景知识的策略是指利用可用的背景知识（例如同义词表、中英文对照词库等）发现元素之间的映射关系，对于上述三种情况设计了三种背景知识的利用策略，分别为保存转换结果、保存已有映射结果、保存领域内中文同义词。基于推理的策略利用关系数据库及本体的语义关系进行推理，设计了两条推理规则，利用关系与属性、类与属性以及属性与数据类型等约束，发现错误的映射，并进行滤除。

[0049] 最后，开发相应的人机界面，半自动地完成关系数据库模式与本体间的映射。设计了一种可视化的、界面友好的映射工具，辅助用户建立映射，用户能够对系统自动建立的映射进行编辑（添加或删除）。系统还提供了一种高效的、问答式的工作模式以便排除更多的错误映射，每次提问都选择最有可能为错误的映射，这样，通过多次提问，最终获得可信度高的映射结果。

[0050] 本发明的船舶装备维修保障信息关系数据库模式与本体间映射方法的具体步骤如图1所示，它包括如下步骤：

[0051] 步骤1：对船舶装备维修保障信息本体的元素和船舶装备维修保障信息关系数据库模式的元素进行分类，将上述船舶装备维修保障信息关系数据库模式的元素中的关系R分为实体类型关系ER和关系类型关系RR，将船舶装备维修保障信息关系数据库模式中的属性A按数据库逆向工程方式分为外键属性FKA和非外键属性NFKA，将船舶装备维修保障信息本体的元素按船舶装备维修保障信息本体自带的说明规范，分为船舶装备维修保障信息本体元素的类C和属性P，将属性P进一步分为船舶装备维修保障信息本体元素的对象属性P_o和船舶装备维修保障信息本体元素的数据类型属性P_d，一般而言，一个实体类型关系应该映射到本体的一个类，一个关系类型关系应该映射到本体的一个对象属性，类似的，外键属性映射到本体的对象属性，而非外键属性可以映射到本体的对象属性，也可以映射到数据

类型属性(需要注意的是,关系类型关系中所有主键且作为外键的属性,不参与到映射过程,否则会导致重复)。按照上述启发式规则,建立船舶装备维修保障信息关系数据库模式和船舶装备维修保障信息本体之间的如下映射类型:

- [0052] 实体型关系ER至船舶装备维修保障信息本体元素的类C的映射类型,即 $\{ER\} \rightarrow \{C\}$;
- [0053] 关系类型关系RR至船舶装备维修保障信息本体元素的对象属性 P_o 的映射类型,即 $\{RR\} \rightarrow \{P_o\}$;
- [0054] 外键属性FKA至船舶装备维修保障信息本体元素的对象属性 P_o 的映射类型,即 $\{FKA\} \rightarrow \{P_o\}$;
- [0055] 非外键属性NFKA至船舶装备维修保障信息本体元素的对象属性 P_o 与船舶装备维修保障信息本体元素的数据类型属性 P_d 并集的映射类型,即 $\{NFKA\} \rightarrow \{P_o \cup P_d\}$;
- [0056] 后续建立的关系数据库模式与本体元素间的映射,必须属于上述四种映射类型的范畴,否则为错误映射;
- [0057] 步骤2:根据如下步骤201~步骤203得到待处理船舶装备维修保障信息关系数据库模式与待处理船舶装备维修保障信息本体(以船舶装备维修保障故障信息为例,建立的本体如图2所示)之间的映射关系:
- [0058] 步骤201:将待处理船舶装备维修保障信息本体的元素的命名方式统一为英文描述,然后判断船舶装备维修保障信息关系数据库模式的元素的命名方式为以下两种命名方式的哪一种:
 - [0059] 第一种船舶装备维修保障信息关系数据库模式的元素的命名方式为:使用英文单词或缩写进行描述,例如“故障信息表”用“fault”、“faultInfor”表示,“故障发生时间”用“time”、“fault_time”表示;
 - [0060] 第二种船舶装备维修保障信息关系数据库模式的元素的命名方式为:直接使用中文进行描述,例如“人员信息表”用“人员信息”、“人员”表示;
- [0061] 步骤202:对于第二种命名方式,首先将对应的船舶装备维修保障信息关系数据库模式的元素转换成英文描述(通过MicrosoftTranslator软件进行翻译),此时即将第二种命名方式转换成了第一种命名方式,然后为待处理船舶装备维修保障信息本体的每个元素和待处理船舶装备维修保障信息关系数据库模式的每个元素分别构建对应的虚拟文档,每个虚拟文档包括上述待处理船舶装备维修保障信息本体的每个元素或船舶装备维修保障信息关系数据库模式的每个元素对应的英文描述,还包括与上述每个元素结构上相邻的元素对应的英文描述(为了引入结构特征来捕获语义信息),根据步骤1的分类方式为不同类型船舶装备维修保障信息本体的元素和船舶装备维修保障信息关系数据库模式的元素构建不同的虚拟文档;对于第一种和第二种情况,为了防止每次都要进行语言转换,将每次转换的结果保存在文本文件中。当面临新的映射任务时,首先对该文本文件进行检索,若存在对应的转换,则直接使用。若不存在,则先进行转换然后将其存入文本文件的末尾,以便下次使用;对于第二种情况,为了防止每次进行相似度计算,对已有的映射结果进行分析,建立舰船装备维修保障领域中文同义词词库,存入文本文件中(正反方向各存一次),当面临新的映射任务时,检索该文本文件,找出元素词汇的同义词,以便更高效地发现映射;
- [0062] 为待处理船舶装备维修保障信息关系数据库模式的元素中的关系,根据以下方式

构建虚拟文档：

[0063] 对于待处理船舶装备维修保障信息关系数据库模式中的关系R,若为实体类型关系ER,则该关系R的虚拟文档只包括自身的英文描述;若为关系类型关系RR,则该关系R的虚拟文档不仅包括其自身的英文描述,还包括关系类型关系RR所引用关系的英文描述;例如图3中关系类型关系T_require的虚拟文档为{“require”、“fault”、“resource”};

[0064] 用vdoc()函数表示虚拟文档,用desc()函数表示待处理船舶装备维修保障信息本体的每个元素或船舶装备维修保障信息关系数据库模式的每个元素的英文描述,具体形式化如下:

[0065]

$$vdoc(R) = \begin{cases} desc(R) & R \in ER \\ desc(R) + desc(rel(A'_s)) & R \in RR, A'_s \in ref(A_s), A_s \in pk(R) \end{cases} \quad (1)$$

[0066] 其中, $A_R \in pk(R)$ 表示 A_R 属于关系R的主键, $A_{R'} \in ref(A_R)$ 表示关系R的主键引用的属性, $rel(A_{R'})$ 表示属性 $A_{R'}$ 所属的关系, $desc(rel(A_{R'}))$ 表示属性 $A_{R'}$ 所属的关系的英文描述, $vdoc(R)$ 表示关系R的虚拟文档, $desc(R)$ 表示关系R的英文描述;

[0067] 对于待处理船舶装备维修保障信息关系数据库模式中的属性A,则其虚拟文档除了自身的英文描述外,还包括该属性A所属的关系的英文描述,如果它是外键属性FKA,则还需进一步考虑该属性A引用的属性所属关系的描述;如果该属性A是非外键属性NFKA,则需补充其数据类型,具体如下:

[0068]

$$vdoc(A) = \begin{cases} desc(A) + desc(rel(A)) + desc(rel(A')) & A' \in ref(A), A \in FKA \\ desc(A) + desc(rel(A)) + desc(type(A)) & A \in NFKA \end{cases} \quad (2)$$

[0069] 其中, $type(A)$ 表示属性A的数据类型, $desc(type(A))$ 表示属性A数据类型的英文描述, $desc(rel(A))$ 表示属性A所属的关系的英文描述, $desc(A)$ 表示属性A的英文描述, $desc(rel(A'))$ 表示属性 A' 所属关系的英文描述;

[0070] 对于船舶装备维修保障信息本体元素的类C,它的虚拟文档就是其自身的英文描述,即:

[0071] $vdoc(C) = desc(C) \quad (3)$

[0072] 对于待处理船舶装备维修保障信息本体中的属性P,则其虚拟文档除了自身的自然语言描述外,还包括它的定义域d(P)和值域r(P)的自然语言描述,对于船舶装备维修保障信息本体元素的数据类型属性P_d,其值域为数据类型,具体如下:

[0073] $vdoc(P) = desc(P) + desc(d(P)) + desc(r(P)) \quad (4)$

[0074] 其中,待处理船舶装备维修保障信息本体中的属性P=船舶装备维修保障信息本体元素的数据类型属性P_d+船舶装备维修保障信息本体元素的对象属性P_o, $vdoc(P)$ 表示待处理船舶装备维修保障信息本体中属性P的虚拟文档, $desc(P)$ 表示待处理船舶装备维修保障信息本体中属性P的英文描述, $desc(d(P))$ 表示定义域d(P)的英文描述, $desc(r(P))$ 表示值域r(P)的英文描述;

[0075] 这样,就为船舶装备维修保障信息关系数据库模式和本体的每个元素构建了虚拟文档,每个虚拟文档都看成向量空间模型(VectorSpace Model,VSM)中的一个向量,对于每

个虚拟文档,其中任一单词i的权重用如下TF-IDF算法进行计算:

$$[0076] \quad tf_i = \frac{n_i}{\sum_k n_k} \quad (5)$$

$$[0077] \quad idf_i = \log \frac{N}{n+1} \quad (6)$$

$$[0078] \quad w_i = tf_i \cdot idf_i \quad (7)$$

[0079] 其中 tf_i 为单词i的词频, n_i 为单词i在该虚拟文档内出现的次数, $\sum_k n_k$ 为该虚拟文档的总词数; idf_i 为逆文档频率,N为虚拟文档总数,n为含有单词i的虚拟文档数, w_i 表示单词i的权重,这样虚拟文档就转换成带权值的单词集合即向量,对于任意两个带权值的单词集合,即向量 e_i 和 e_j 之间的相似度 $sim(e_i, e_j)$ 用如下余弦相似度公式来衡量:

$$[0080] \quad sim(e_i, e_j) = \frac{\sum_{k=1}^D w_{ik} w_{jk}}{\sqrt{\sum_{k=1}^D w_{ik}^2} \sqrt{\sum_{k=1}^D w_{jk}^2}} \quad (8)$$

[0081] 其中, w_{ik} 表示向量 e_i 中单词k的权值, w_{jk} 表示向量 e_j 中单词k的权值,D为向量的维度,即所有单词的总数;

[0082] 步骤203:按照步骤1建立的映射类型,和步骤201、步骤202中的方法,计算船舶装备维修保障信息关系数据库模式中元素 e_s 与本体中元素 e_t 之间的相似度 $sim(e_s, e_t)$,即船舶装备维修保障信息关系数据库模式的元素对应的向量之间的相似度,得到所有满足映射类型要求的元素间相似度后,按照相似度从大到小排序,设定阈值 θ ($\theta \in (0, 1)$),过滤掉所有相似度小于 θ 的元素对,对于相似度大于或等于 θ 的元素对,建立这两个元素之间的映射关系,即得到船舶装备维修保障信息关系数据库模式与本体间的映射。

[0083] 上述步骤2在实施时,为了提高效率,利用Apache Lucene搜索引擎实现该方法。Lucene首先分别提取关系数据库模式元素和本体元素的词汇,进行标准的预处理(全部转换成小写、去掉停用词、正规化),并为每个元素的虚拟文档建立索引(Index)。然后为关系数据库的每个元素建立查询语句(对应的虚拟文档),查询本体的索引,将返回结果(Hits)中相似度最大的Top-K个结果保存。然后将关系数据库与本体互换,重复上一步。最后计算上述两个步骤的交集。这样不仅效率更高,而且保证了查全率。

[0084] 上述步骤203后还包括步骤3:

[0085] 为了发现步骤203中得到的船舶装备维修保障信息关系数据库模式与本体间的映射是否存在错误,并将存在的错误排除,设计了基于推理的策略(关系数据库模式中关系的属性必须依靠关系存在,并且属性具有数据类型等诸多约束,因此可以利用这些信息进行一些简单的推理,发现错误的映射,得到可信度更高的映射结果),主要包括如下两条推理规则:

[0086] 如果存在待处理船舶装备维修保障信息关系数据库模式中的属性A与待处理船舶

装备维修保障信息本体中的属性P之间的映射m,但不存在所述属性A所属的船舶装备维修保障信息关系数据库模式的元素中的关系R与所述属性P定义域所属的类C之间的映射,则映射m为错误映射;

[0087] 如果存在待处理船舶装备维修保障信息关系数据库模式中的属性A与船舶装备维修保障信息本体元素的数据类型属性Pa之间的映射m₁,但是所述属性A的数据类型与所述数据类型属性Pa的值域所属的数据类型为预设的不兼容数据类型(例如整数型int与长整数型long兼容,与字符串型string不兼容;单精度浮点型float与双精度浮点型double兼容,与布尔型boolean不兼容;时间型time与日期时间型datetime兼容,与字节型byte不兼容等),则m₁为错误映射;

[0088] 在船舶装备维修保障信息关系数据库模式与本体间的映射结果中删除上述错误映射m和错误映射m₁。

[0089] 上述技术方案的步骤2中,对船舶维修保障领域本体的建立和船舶维修保障领域关系数据库元素的命名规则进行了分析。在此前提下,设计了多种策略发现二者之间的映射关系,包括:基于语言学的策略、基于背景知识的策略和基于推理的策略。先实施语言学策略,过程中可以利用其他背景知识,最后利用基于推理的策略过滤错误的映射,自动化地建立关系数据库模式与本体元素的映射关系。

[0090] 上述步骤3后还包括步骤4:

[0091] 为了进一步提高上述得到的删除错误映射m和错误映射m₁后的船舶装备维修保障信息关系数据库模式与本体间映射的准确率和查全率,设计了如下系统提问、用户回答的交互方式来排除更多错误映射,从而提高映射结果的质量,如果逐条提问,则当关系数据库模式和本体的规模较大时,比较耗时且容易出错。因此,在该问答过程中,为了获取最具信息量的映射来问询用户,提问规则为:

[0092] 规则一、如果存在船舶装备维修保障信息关系数据库模式和本体间映射(e_s,e_t)和映射(e_{s'},e_{t'}),且其中e_s=e_{s'}或e_t=e_{t'},e_s、e_{s'}为船舶装备维修保障信息关系数据库模式中的元素,e_t、e_{t'}为船舶装备维修保障信息本体的元素,则映射(e_s,e_t)和映射(e_{s'},e_{t'})中有一个映射为错误映射,选择相似度差值最小的映射,即

[0093] $Q(e_s, e_t) = \min \{ |sim(e_s, e_t) - sim(e_s', e_t')| \}; e_s = e_s' \text{ 或 } e_t = e_t'$

[0094] 规则二、选择相似度sim(e_s,e_t)与阈值θ相差最小的映射,即

[0095] $Q(e_s, e_t) = \min \{ |sim(e_s, e_t) - \theta| \}$

[0096] 该方式通过不断迭代,每次迭代选择满足上述两条规则的映射,用户对映射正确与否做出判断,使得系统能够在最少问答次数的情况下,获得最大的映射质量增加。

[0097] 除此之外,还开发了可视化的映射工具,能够辅助领域专家建立关系数据库模式与全局本体的映射关系,如图4所示。利用OWL(Web Ontology Language,网络本体语言)API(应用程序编程接口)解析本体文件,利用相应的数据库访问接口(例如JDBC(Java DataBase Connectivity,java数据库连接)等)获取关系数据库元素,以树状结构显示关系数据库模式的关系和属性,以及本体的类及数据类型属性,而对象属性则以隐性信息展现。映射结果以列表形式展示,用户可以对映射结果进行编辑(添加、删除并保存)。

[0098] 本说明书未作详细描述的内容属于本领域专业技术人员公知的现有技术。

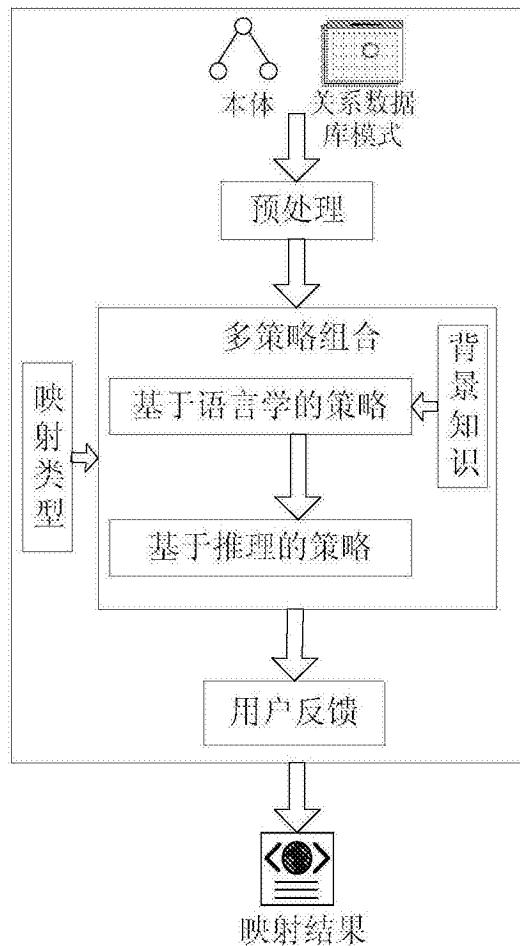


图1

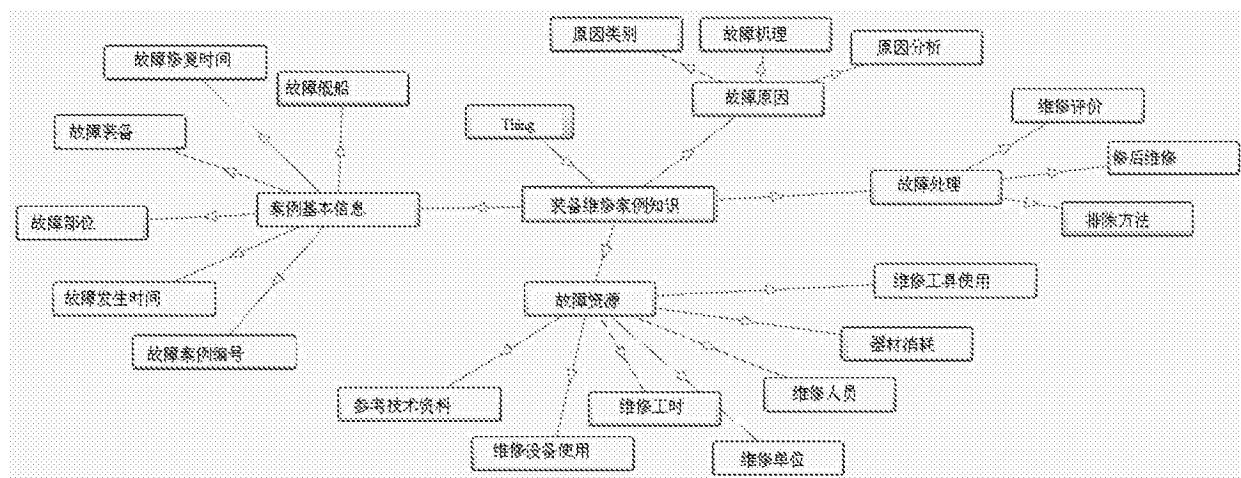


图2

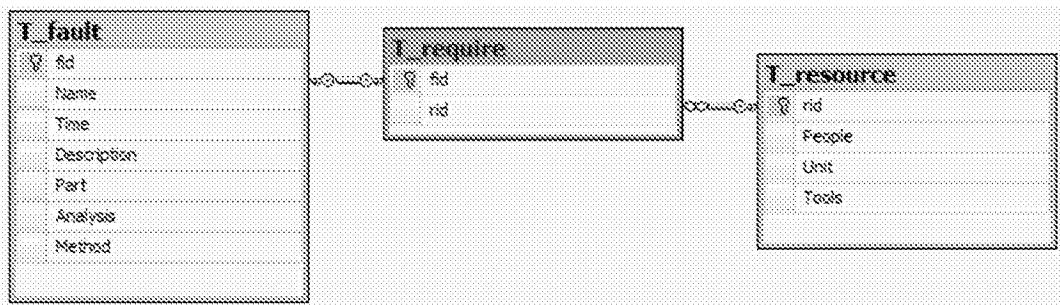


图3

| | | 关系矩阵 | | | |
|---------|-------|-------------|-------|------|------|
| | | 源元素 | 目标元素 | 属性 | 操作关系 |
| Fault | 1 | Y008 | 故障及失败 | 0..1 | * |
| | 2 | Time | 故障及失败 | 0..1 | * |
| | 3 | Description | 故障及失败 | 0..1 | * |
| | 4 | Part | 故障及失败 | 0..1 | * |
| | 5 | Analysis | 故障及失败 | 0..1 | * |
| | 6 | Method | 故障及失败 | 0..1 | * |
| | 7 | People | 故障人员 | 0..1 | * |
| | 8 | Unit | 故障单位 | 0..1 | * |
| | 9 | Tools | 故障工具 | 0..1 | * |
| Feature | 10 | Y008 | 故障及失败 | 0..1 | * |
| | 11 | Time | 故障及失败 | 0..1 | * |
| | 12 | Description | 故障及失败 | 0..1 | * |
| | 13 | Part | 故障及失败 | 0..1 | * |
| | 14 | Analysis | 故障及失败 | 0..1 | * |
| | 15 | Method | 故障及失败 | 0..1 | * |
| | 16 | People | 故障人员 | 0..1 | * |
| | 17 | Unit | 故障单位 | 0..1 | * |
| | 18 | Tools | 故障工具 | 0..1 | * |
| | 19 | Y008s | 故障及失败 | 0..1 | * |
| | 20 | Y008t | 故障及失败 | 0..1 | * |
| | 21 | Y008u | 故障及失败 | 0..1 | * |
| | 22 | Y008v | 故障及失败 | 0..1 | * |
| | 23 | Y008w | 故障及失败 | 0..1 | * |
| | 24 | Y008x | 故障及失败 | 0..1 | * |
| 25 | Y008y | 故障及失败 | 0..1 | * | |
| 26 | Y008z | 故障及失败 | 0..1 | * | |
| 27 | Y008s | 故障及失败 | 0..1 | * | |
| 28 | Y008t | 故障及失败 | 0..1 | * | |
| 29 | Y008u | 故障及失败 | 0..1 | * | |
| 30 | Y008v | 故障及失败 | 0..1 | * | |
| 31 | Y008w | 故障及失败 | 0..1 | * | |
| 32 | Y008x | 故障及失败 | 0..1 | * | |
| 33 | Y008y | 故障及失败 | 0..1 | * | |
| 34 | Y008z | 故障及失败 | 0..1 | * | |
| 35 | Y008s | 故障及失败 | 0..1 | * | |
| 36 | Y008t | 故障及失败 | 0..1 | * | |
| 37 | Y008u | 故障及失败 | 0..1 | * | |
| 38 | Y008v | 故障及失败 | 0..1 | * | |
| 39 | Y008w | 故障及失败 | 0..1 | * | |
| 40 | Y008x | 故障及失败 | 0..1 | * | |
| 41 | Y008y | 故障及失败 | 0..1 | * | |
| 42 | Y008z | 故障及失败 | 0..1 | * | |
| 43 | Y008s | 故障及失败 | 0..1 | * | |
| 44 | Y008t | 故障及失败 | 0..1 | * | |
| 45 | Y008u | 故障及失败 | 0..1 | * | |
| 46 | Y008v | 故障及失败 | 0..1 | * | |
| 47 | Y008w | 故障及失败 | 0..1 | * | |
| 48 | Y008x | 故障及失败 | 0..1 | * | |
| 49 | Y008y | 故障及失败 | 0..1 | * | |
| 50 | Y008z | 故障及失败 | 0..1 | * | |
| 51 | Y008s | 故障及失败 | 0..1 | * | |
| 52 | Y008t | 故障及失败 | 0..1 | * | |
| 53 | Y008u | 故障及失败 | 0..1 | * | |
| 54 | Y008v | 故障及失败 | 0..1 | * | |
| 55 | Y008w | 故障及失败 | 0..1 | * | |
| 56 | Y008x | 故障及失败 | 0..1 | * | |
| 57 | Y008y | 故障及失败 | 0..1 | * | |
| 58 | Y008z | 故障及失败 | 0..1 | * | |
| 59 | Y008s | 故障及失败 | 0..1 | * | |
| 60 | Y008t | 故障及失败 | 0..1 | * | |
| 61 | Y008u | 故障及失败 | 0..1 | * | |
| 62 | Y008v | 故障及失败 | 0..1 | * | |
| 63 | Y008w | 故障及失败 | 0..1 | * | |
| 64 | Y008x | 故障及失败 | 0..1 | * | |
| 65 | Y008y | 故障及失败 | 0..1 | * | |
| 66 | Y008z | 故障及失败 | 0..1 | * | |
| 67 | Y008s | 故障及失败 | 0..1 | * | |
| 68 | Y008t | 故障及失败 | 0..1 | * | |
| 69 | Y008u | 故障及失败 | 0..1 | * | |
| 70 | Y008v | 故障及失败 | 0..1 | * | |
| 71 | Y008w | 故障及失败 | 0..1 | * | |
| 72 | Y008x | 故障及失败 | 0..1 | * | |
| 73 | Y008y | 故障及失败 | 0..1 | * | |
| 74 | Y008z | 故障及失败 | 0..1 | * | |
| 75 | Y008s | 故障及失败 | 0..1 | * | |
| 76 | Y008t | 故障及失败 | 0..1 | * | |
| 77 | Y008u | 故障及失败 | 0..1 | * | |
| 78 | Y008v | 故障及失败 | 0..1 | * | |
| 79 | Y008w | 故障及失败 | 0..1 | * | |
| 80 | Y008x | 故障及失败 | 0..1 | * | |
| 81 | Y008y | 故障及失败 | 0..1 | * | |
| 82 | Y008z | 故障及失败 | 0..1 | * | |
| 83 | Y008s | 故障及失败 | 0..1 | * | |
| 84 | Y008t | 故障及失败 | 0..1 | * | |
| 85 | Y008u | 故障及失败 | 0..1 | * | |
| 86 | Y008v | 故障及失败 | 0..1 | * | |
| 87 | Y008w | 故障及失败 | 0..1 | * | |
| 88 | Y008x | 故障及失败 | 0..1 | * | |
| 89 | Y008y | 故障及失败 | 0..1 | * | |
| 90 | Y008z | 故障及失败 | 0..1 | * | |
| 91 | Y008s | 故障及失败 | 0..1 | * | |
| 92 | Y008t | 故障及失败 | 0..1 | * | |
| 93 | Y008u | 故障及失败 | 0..1 | * | |
| 94 | Y008v | 故障及失败 | 0..1 | * | |
| 95 | Y008w | 故障及失败 | 0..1 | * | |
| 96 | Y008x | 故障及失败 | 0..1 | * | |
| 97 | Y008y | 故障及失败 | 0..1 | * | |
| 98 | Y008z | 故障及失败 | 0..1 | * | |
| 99 | Y008s | 故障及失败 | 0..1 | * | |
| 100 | Y008t | 故障及失败 | 0..1 | * | |
| 101 | Y008u | 故障及失败 | 0..1 | * | |
| 102 | Y008v | 故障及失败 | 0..1 | * | |
| 103 | Y008w | 故障及失败 | 0..1 | * | |
| 104 | Y008x | 故障及失败 | 0..1 | * | |
| 105 | Y008y | 故障及失败 | 0..1 | * | |
| 106 | Y008z | 故障及失败 | 0..1 | * | |
| 107 | Y008s | 故障及失败 | 0..1 | * | |
| 108 | Y008t | 故障及失败 | 0..1 | * | |
| 109 | Y008u | 故障及失败 | 0..1 | * | |
| 110 | Y008v | 故障及失败 | 0..1 | * | |
| 111 | Y008w | 故障及失败 | 0..1 | * | |
| 112 | Y008x | 故障及失败 | 0..1 | * | |
| 113 | Y008y | 故障及失败 | 0..1 | * | |
| 114 | Y008z | 故障及失败 | 0..1 | * | |
| 115 | Y008s | 故障及失败 | 0..1 | * | |
| 116 | Y008t | 故障及失败 | 0..1 | * | |
| 117 | Y008u | 故障及失败 | 0..1 | * | |
| 118 | Y008v | 故障及失败 | 0..1 | * | |
| 119 | Y008w | 故障及失败 | 0..1 | * | |
| 120 | Y008x | 故障及失败 | 0..1 | * | |
| 121 | Y008y | 故障及失败 | 0..1 | * | |
| 122 | Y008z | 故障及失败 | 0..1 | * | |
| 123 | Y008s | 故障及失败 | 0..1 | * | |
| 124 | Y008t | 故障及失败 | 0..1 | * | |
| 125 | Y008u | 故障及失败 | 0..1 | * | |
| 126 | Y008v | 故障及失败 | 0..1 | * | |
| 127 | Y008w | 故障及失败 | 0..1 | * | |
| 128 | Y008x | 故障及失败 | 0..1 | * | |
| 129 | Y008y | 故障及失败 | 0..1 | * | |
| 130 | Y008z | 故障及失败 | 0..1 | * | |
| 131 | Y008s | 故障及失败 | 0..1 | * | |
| 132 | Y008t | 故障及失败 | 0..1 | * | |
| 133 | Y008u | 故障及失败 | 0..1 | * | |
| 134 | Y008v | 故障及失败 | 0..1 | * | |
| 135 | Y008w | 故障及失败 | 0..1 | * | |
| 136 | Y008x | 故障及失败 | 0..1 | * | |
| 137 | Y008y | 故障及失败 | 0..1 | * | |
| 138 | Y008z | 故障及失败 | 0..1 | * | |
| 139 | Y008s | 故障及失败 | 0..1 | * | |
| 140 | Y008t | 故障及失败 | 0..1 | * | |
| 141 | Y008u | 故障及失败 | 0..1 | * | |
| 142 | Y008v | 故障及失败 | 0..1 | * | |
| 143 | Y008w | 故障及失败 | 0..1 | * | |
| 144 | Y008x | 故障及失败 | 0..1 | * | |
| 145 | Y008y | 故障及失败 | 0..1 | * | |
| 146 | Y008z | 故障及失败 | 0..1 | * | |
| 147 | Y008s | 故障及失败 | 0..1 | * | |
| 148 | Y008t | 故障及失败 | 0..1 | * | |
| 149 | Y008u | 故障及失败 | 0..1 | * | |
| 150 | Y008v | 故障及失败 | 0..1 | * | |
| 151 | Y008w | 故障及失败 | 0..1 | * | |
| 152 | Y008x | 故障及失败 | 0..1 | * | |
| 153 | Y008y | 故障及失败 | 0..1 | * | |
| 154 | Y008z | 故障及失败 | 0..1 | * | |
| 155 | Y008s | 故障及失败 | 0..1 | * | |
| 156 | Y008t | 故障及失败 | 0..1 | * | |
| 157 | Y008u | 故障及失败 | 0..1 | * | |
| 158 | Y008v | 故障及失败 | 0..1 | * | |
| 159 | Y008w | 故障及失败 | 0..1 | * | |
| 160 | Y008x | 故障及失败 | 0..1 | * | |
| 161 | Y008y | 故障及失败 | 0..1 | * | |
| 162 | Y008z | 故障及失败 | 0..1 | * | |
| 163 | Y008s | 故障及失败 | 0..1 | * | |
| 164 | Y008t | 故障及失败 | 0..1 | * | |
| 165 | Y008u | 故障及失败 | 0..1 | * | |
| 166 | Y008v | 故障及失败 | 0..1 | * | |
| 167 | Y008w | 故障及失败 | 0..1 | * | |
| 168 | Y008x | 故障及失败 | 0..1 | * | |
| 169 | Y008y | 故障及失败 | 0..1 | * | |
| 170 | Y008z | 故障及失败 | 0..1 | * | |
| 171 | Y008s | 故障及失败 | 0..1 | * | |
| 172 | Y008t | 故障及失败 | 0..1 | * | |
| 173 | Y008u | 故障及失败 | 0..1 | * | |
| 174 | Y008v | 故障及失败 | 0..1 | * | |
| 175 | Y008w | 故障及失败 | 0..1 | * | |
| 176 | Y008x | 故障及失败 | 0..1 | * | |
| 177 | Y008y | 故障及失败 | 0..1 | * | |
| 178 | Y008z | 故障及失败 | 0..1 | * | |
| 179 | Y008s | 故障及失败 | 0..1 | * | |
| 180 | Y008t | 故障及失败 | 0..1 | * | |
| 181 | Y008u | 故障及失败 | 0..1 | * | |
| 182 | Y008v | 故障及失败 | 0..1 | * | |
| 183 | Y008w | 故障及失败 | 0..1 | * | |
| 184 | Y008x | 故障及失败 | 0..1 | * | |
| 185 | Y008y | 故障及失败 | 0..1 | * | |
| 186 | Y008z | 故障及失败 | 0..1 | * | |
| 187 | Y008s | 故障及失败 | 0..1 | * | |
| 188 | Y008t | 故障及失败 | 0..1 | * | |
| 189 | Y008u | 故障及失败 | 0..1 | * | |
| 190 | Y008v | 故障及失败 | 0..1 | * | |
| 191 | Y008w | 故障及失败 | 0..1 | * | |
| 192 | Y008x | 故障及失败 | 0..1 | * | |
| 193 | Y008y | 故障及失败 | 0..1 | * | |
| 194 | Y008z | 故障及失败 | 0..1 | * | |
| 195 | Y008s | 故障及失败 | 0..1 | * | |
| 196 | Y008t | 故障及失败 | 0..1 | * | |
| 197 | Y008u | 故障及失败 | 0..1 | * | |
| 198 | Y008v | 故障及失败 | 0..1 | * | |
| 199 | Y008w | 故障及失败 | 0..1 | * | |
| 200 | Y008x | 故障及失败 | 0..1 | * | |
| 201 | Y008y | 故障及失败 | 0..1 | * | |
| 202 | Y008z | 故障及失败 | 0..1 | * | |
| 203 | Y008s | 故障及失败 | 0..1 | * | |
| 204 | Y008t | 故障及失败 | 0..1 | * | |
| 205 | Y008u | 故障及失败 | 0..1 | * | |
| 206 | Y008v | 故障及失败 | 0..1 | * | |
| 207 | Y008w | 故障及失败 | 0..1 | * | |
| 208 | Y008x | 故障及失败 | 0..1 | * | |
| 209 | Y008y | 故障及失败 | 0..1 | * | |
| 210 | Y008z | 故障及失败 | 0..1 | * | |
| 211 | Y008s | 故障及失败 | 0..1 | * | |
| 212 | Y008t | 故障及失败 | 0..1 | * | |
| 213 | Y008u | 故障及失败 | 0..1 | * | |
| 214 | Y008v | 故障及失败 | 0..1 | * | |
| 215 | Y008w | 故障及失败 | 0..1 | * | |
| 216 | Y008x | 故障及失败 | 0..1 | * | |
| 217 | Y008y | 故障及失败 | 0..1 | * | |
| 218 | Y008z | 故障及失败 | 0..1 | * | |
| 219 | Y008s | 故障及失败 | 0..1 | * | |
| 220 | Y008t | 故障及失败 | 0..1 | * | |
| 221 | Y008u | 故障及失败 | 0..1 | * | |
| 222 | Y008v | 故障及失败 | 0..1 | * | |
| 223 | Y008w | 故障及失败 | 0..1 | * | |
| 224 | Y008x | 故障及失败 | 0..1 | * | |
| 225 | Y008y | 故障及失败 | 0..1 | * | |
| 226 | Y008z | 故障及失败 | 0..1 | * | |
| 227 | Y008s | 故障及失败 | 0..1 | * | |
| 228 | Y008t | 故障及失败 | 0..1 | * | |
| 229 | Y008u | 故障及失败 | 0..1 | * | |
| 230 | Y008v | 故障及失败 | 0..1 | * | |
| 231 | Y008w | 故障及失败 | 0..1 | * | |
| 232 | Y008x | 故障及失败 | 0..1 | * | |
| 233 | Y008y | 故障及失败 | 0..1 | * | |
| 234 | Y008z | 故障及失败 | 0..1 | * | |
| 235 | Y008s | 故障及失败 | 0..1 | * | |
| 236 | Y008t | 故障及失败 | 0..1 | * | |
| 237 | Y008u | 故障及失败 | 0..1 | * | |
| 238 | Y008v | 故障及失败 | 0..1 | * | |
| 239 | Y008w | 故障及失败 | 0..1 | * | |
| 240 | Y008x | 故障及失败 | 0..1 | * | |
| 241 | Y008y | 故障及失败 | 0..1 | * | |
| 242 | Y008z | 故障及失败 | 0..1 | * | |
| 243 | Y008s | 故障及失败 | 0..1 | * | |
| 244 | Y008t | 故障及失败 | 0..1 | * | |
| 245 | Y008u | 故障及失败 | 0..1 | * | |
| 246 | Y008v | 故障及失败 | 0..1 | * | |
| 247 | Y008w | 故障及失败 | 0..1 | * | |
| 248 | Y008x | 故障及失败 | 0..1 | * | |
| 249 | Y008y | 故障及失败 | 0..1 | * | |
| 250 | Y008z | 故障及失败 | 0..1 | * | |
| 251 | Y008s | 故障及失败 | 0..1 | * | |
| 252 | Y008t | 故障及失败 | 0..1 | * | |
| 253 | | | | | |