文章编号: 1001- 2486(2008) 06- 0078- 05

EHDM 的 DL 封装机制及在 Z39. 50 协议上的应用*

王 博, 郭 波
(国防科技大学 信息系统与管理学院, 湖南 长沙 410073)

摘 要: Z39. 50 被广泛应用在异构数据源的检索中, 由于数据源的动态性和自治性, 无法准确获得其语义, 很难建立 Z39. 50 与数据源的映射关系, 为进行准确的信息检索操作, 提出扩展超图数据模型(EHDM), 对 EHDM 进行层次抽象, 以 EHDM 的两类节点和边作为 Z39. 50 协议的访问点, 通过在 EHDM 和 Z39. 50 之间建立 DL 层, 将 DL 系统的概念、导出概念分别映射到 Z39. 50 协议包装器上, 解决 Z39. 50 协议查询对结构化数据源支持不好、出现错误查询结果集等问题, 定义类 DataSpace 的三层结构实现根据需求进行集成的"量入为出"的数据集成策略。

关键词: EHDM; DL; Z39. 50 协议; 映射
中图分类号: TP311 文献标识码: A

DL Wrapper Mechanism of EHDM with Applications on Z39. 50

WANG Bo, GUO Bo
(College of Information System and Management, National Univ. of Defense Technology, Changsha 410073, China)

Abstract: Z39. 50 is widely used for searching and retrieving information spread over a number of heterogeneous sources, but it is hard to construct the mapping relationships between Z39. 50 and the data sources, since the semantics of the dynamic and autonomy data sources cannot be exactly derived so as to achieve an exact information retrieval operation. This paper proposes the extend hyper graph data model based on HDM, which can be abstracted into levels, taking two kinds of nodes and edges of EHDM as access points of Z39. 50. It can solve the problems of badly supporting of querying on structured data sources and query failures by constructing the DL layer between EHDM and Z39. 50, in which the concepts and derived concepts are mapped onto Z39. 50 wrappers. Thus, defining three layers structure as in dataspace systems can realize the data integration strategy named "integrate when needed".

Key words: EHDM; DL; Z39. 50 protocol; mapping

超图数据模型(HDM)[3] 能够较好地描述各种高层数据模型, 方便地实现异构数据源 BAV[4] 方式的模式集成, 但 HDM 主要描述数据源以及全局模式的静态行为, 其语义信息局限在数据模型的结构表达、模式结构元素命名以及相互关系的规定上, 无法表达内在的语义信息, HDM 模式结构元素强烈依赖于数据源或集成用户的定义, 需要数据源知识才能实现信息检索, 对于高级的、面向一般用户的信息检索并不适合, Z39. 50 协议支持异构数据源上的检索[7], 用于在分布式的环境下信息的搜索和浏览, 协议的基础是数据源的 Z39. 50 协议包装, 关键是非结构化组织的访问点同底层数据源的映射。而 Z39. 50 的访问点方式的数据集成和信息检索对于结构化数据源的支持并不理想, Velegrakis 等在面向对象的语义网数据模型 TELOS 的基础上用描述逻辑对 Z39. 50 协议进行包装, 而面向对象语义网数据模型表达能力不强, 其语义定义具有较大的模糊性, 不能很好地反映模型结构元素之间的语义关系尤其是约束关系, 本体[6] 是解决语义集成的有效方法, 而完善的本体构建是个耗时且无法保证质量的过程, 如果不依赖于严格的数据模型定义, 本体方法实现的语义集成的准确性和推理效率必然受到限制。本文在 HDM 基础上提出扩展超图数据模型, 定义数据模型的动态行为, 以该模型的结构元素作为 Z39. 50 协议访问点, 采用描述逻辑(DL) 定义异构数据源的扩展超图数据模型视图, 并给出推理关系, 定义 DL 查询, 对基于

* 收稿日期: 2008- 06- 15
基金项目: 国家部委资助项目(513190801)
作者简介: 王博(1980—), 男, 博士生。

HDM 的 BAV[1] 方式的查询进行扩展。

1　基本概念

1.1　扩展超图数据模型

定义 1　扩展超图数据模型(Extended Hyper-Graph Data Model, 简记为 EHDM)定义为一个包含边和节点及相关约束的图形结构数据模型, 包括实体和主行为两类节点; 实体属性关系和行为属性及关系两类边, 表示为一个三元组: $\langle Nodes, Edges, Constraints\rangle$, 其中:

• $Nodes \subseteq \{\langle\langle n_n\rangle\rangle \mid n_n \in Names \cup [Name_1, Name_2, \cdots, Name_n]\}$, 为 EHDM 节点的集合, 其中 $Names \subseteq Schemes$, $Node$ 用来表示数据模式中的实体以及行为结构。

• $Edges \subseteq Names \times Seq(Schemes)\ \langle n_0, n_1, \cdots, n_m\rangle \in Schemes$ *if* $m \geqslant 1$, $n_0 \in Names$, *and* $n_i \in Schemes$ *for all* $1 \leqslant i \leqslant m$, 其中 $Schemes = Nodes \cup Edges$, $Edges$ 用来表示实体关系以及行为属性。

• $Constraints \subseteq \{c(s_1, \cdots, s_n) \mid c \in Funcs \wedge s_1 \in Schemes \wedge \cdots \wedge Schemes\}$ 用来定义 EHDM 约束, $Constraints \subseteq BoolQueries$, $BoolQueries$ 为 L 上的布尔值查询, $Constraints$ 是 q 布尔查询集合, 其中 $VARS(q) \subseteq Nodes \cup Edges$。

图 1 上半部分给出了一个装备单元信息的 EHDM, 下半部分给出了其对应的数据实例, 本文的后续部分将以该 EHDM 为示例。

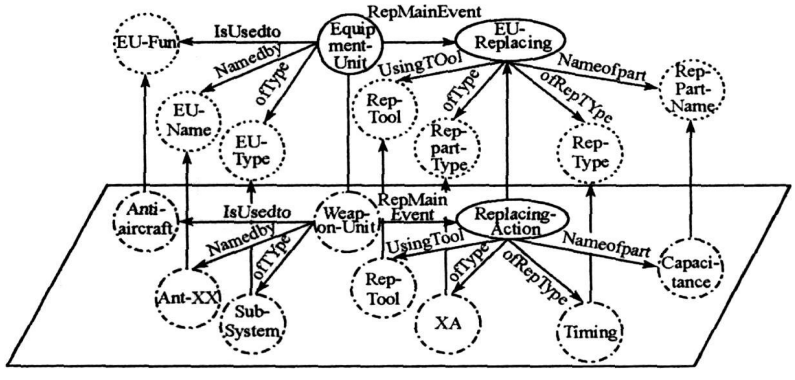

图 1　EHDM 示例

Fig. 1　Example of EHDM

1.2　Z39.50 协议简介

Z39.50 是个面向会话的有状态协议, 基于标准的客户服务器结构, 定义在由客户和服务器共享的公共信息模型上, 由一个域列表组成(Access Points, APs), 查询基于访问点操作, 域词汇集合主要用于底层的数据源的检索, Z39.50 采用布尔连接符、搜索项(属性值对)以及一些限定词(大于, 小于等)来表示复杂查询。为实现互操作, Z39.50 建立在如下基础上:

(1) 用于控制和管理客户和服务器之间通信的标准的消息、格式以及程序;

(2) 维护全局的 AP, 用来对数据源的具有代表性的细节进行抽象, 并作为信息查询和访问的入口;

(3) 基本的文本检索原语, 用来表示具有域—值对格式的布尔查询。

数据源需要根据 Z39.50 规范面向其特定的应用对真实的数据组织、格式以及查询能力进行包装来实现 Z39.50 查询, 一般采用元数据方式。建立数据源和 Z39.50 视图之间的映射关系是实现高质量数据检索的关键。Z39.50 包装器直接建立在异构数据模型之上, 缺乏对结构化数据源的有效支持。本文采用数据空间[2] 准则, 根据数据检索和映射的需求在高层进行映射。Z39.50 协议构建包装器包括三类转换:

(1) Z39.50 APs 到底层数据源的结构和语义的转换;

(2) Z39.50 布尔查询到数据源查询语言的转换;

(3) 原有数据格式到 Z39.50 协议语法格式的转换。

APs 对于非结构化的数据源有很好的支持, 只需建立 APs 到数据源属性或标签的映射即可, 然而对于结构化数据源, 该转换过程十分复杂。主要是由于 Z39.50 协议的信息视图与数据源模型的诸多不匹配, 为解决该问题, 本文以 EHDM 作为数据模型, 支持 TELOS 的动态特性。

在 Z39.50 协议和 EHDM 之间建立中间层, 该层基于描述逻辑(Description Logics, DL)定义, DL 提供

了一种可声明的语言,采用建模原语来对相关联的对象集合进行表示和推理。首先从表示数据源或组织的基本概念和角色开始,以导出概念的方式获取 AP 映射的语义,导出概念由基本的概念和 DL 概念构造器形成。由于 DL 既可以作为知识表示语言也可以作为查询语言,得到的这些表示为视图的概念可以与 Z39.50 协议查询语言对应,但实现对数据源的 EHDM 封装,建立 DL 层以支持 Z39.50 协议主要面临如下问题:

(1) 数据源同 APs 的匹配性:在 Z39.50 协议中,AP 的具体含义定义在 profile 中,随着数据源规模不断扩大、自治性增强,无法准确了解数据源的语义信息,AP 可能仅支持少部分的比较普通的数据源元信息,因此可能经常遇到无结果或者无法执行的查询。一种解决方法是只对支持的 AP 进行查询,但这种方式不能明确告知返回的结果是部分还是全部用户的请求,此外,对于一些包含连接符、比较符以及限定词的查询来说,忽略其中的一部分可能带来意想不到的结果,例如同名异义词汇的查询可能导致完全不符合用户需求的结果,查询准确性差。

(2) APs 与数据源映射的复杂性:映射问题一直是数据集成的核心问题,本文定义为底层 EHDM 与 Z39.50 上层访问点之间的映射,由于以 EHDM 作为数据模型,Z39.50 协议的映射机制必须考虑数据源的动态特性。

2 EHDM 的 DL 封装机制

2.1 EHDM 数据源的 DL 知识表示

DL 提供了对象及其关系的表述和推理能力,且在支持数据集成方面有了扩展,提供了相应的形式化描述能力,用来在大规模的集成视图上进行建模和推理。DL 核心的建模原语包括概念、角色以及个体。概念用来描述领域内同类个体的共同特征,个体定义为现实世界实体的描述,角色描述个体之间的关系。DL 系统的两个基本组件为术语盒(TBox)和断言盒(ABox)。前者包括概念描述,后者为关于个体的一些断言。DL 包括原始和导出两类概念,原始概念指定了个体作为概念一员的必要条件,导出概念给出了充要条件,个体定义为原始概念的一个成员,导出个体可以由 DL 系统推导得到。DL 知识基解释 Σ 定义为 $\Gamma=(\Gamma(\Delta),\Gamma(\cdot))$,其中 $\Gamma(\Delta)$ 表示非空个体集合,$\Gamma(\cdot)$ 为解释函数,建立了概念到 $\Gamma(\Delta)$ 子集以及角色到 $\Gamma(\Delta)\times\Gamma(\Delta)$ 子集的映射关系。

概念 C 的一个解释(表示为 $\Gamma(C)$)定义为个体的集合,概念 C 的实例定义为个体或个体的导出形式。一个概念 C_1 被概念 C_2 包含(表示为 $C_1 \prec C_2$)当且仅当对于所有解释,满足 $\Gamma(C_1)\subseteq\Gamma(C_2)$。基于该包含关系,一个概念集合可以形成一个包括概念下界 $\underline{r}$ 和概念上界 $\bar{r}$ 的分类。$\underline{r}$ 定义了满足 $\Gamma(\underline{r})\equiv\phi$ 的概念,$\bar{r}$ 为满足 $\Gamma(\bar{r})\equiv\Delta$ 的概念。

表 1 DL 基本概念操作及其含义

Tab. 1 Operators and meanings of DL concepts

名称	表达式	定义
概念名称	A	$\Gamma(A)$
概念下界	$\underline{r}$	ϕ
概念上界	$\bar{r}$	Δ
并	$A\cup C$	$\{i\mid i\in\Gamma(A)\cup\Gamma(C)\}$
交	$A\cap C$	$\{i\mid i\in\Gamma(A)\cap\Gamma(C)\}$
存在约束	$\exists R.C$	$\{i_1\mid\exists i_2:(i_1,i_2)\in\Gamma(R)\wedge i_2\in\Gamma(C)\}$
全称约束	$\forall R.C$	$\{i_1\mid\forall i_2:(i_1,i_2)\in\Gamma(R)\rightarrow i_2\in\Gamma(C)\}$
非	$\neg A$	$\{i\mid i\notin\Gamma(A)\}$
角色名称	R	$\Gamma(R)$
角色关系	$_A\mid R\mid_B$	$\{(i_1,i_2)\mid(i_1,i_2)\in\Gamma(R)\cap(\Gamma(A)\times\Gamma(B))\}$
反身	R^{-1}	$\{(i_1,i_2)\mid(i_2,i_1)\in\Gamma(R)\}$

TBox 定义为一个有限公理的集合,具有如下形式的一种:$A\prec D$(即 $\Gamma(A)\subseteq\Gamma(D)$),$_C\mid R\mid_D$ 以及具有形如 $K\cdot E$ 概念定义的有限集合,其中 A, C, D 为原始概念,K 为导出概念,R 为角色,E 为从其他概念中获取的概念,表 1 给出了概念 E 的构造方式,更复杂的概念 E 可以由基础构造算子的复合运算得到。TBox 中不相交的基本概念以 $A\parallel C(\Gamma(A)\cap\Gamma(C)\equiv\phi)$ 的形式给出。ABox 定义在有限的声明集合基础上,形

式为 $C(a)$ 或 $R(a,b)$, 前者表示个体 a 属于概念 C, 后者表示存在一个从个体 a 到 b 的一个角色 R, 满足 a 和 b 分别属于 $\exists R\exists_D$ 的两个概念域 C 和 D。DL 系统 Σ 的基本推理服务包括:

- 概念可满足性($\Sigma \not\models C \equiv \underline{\perp}$): 用来检验一个概念是否有非空解释;
- 包含性检验($\Sigma \models C_1 \prec C_2$): 用来检测概念 C_2 是否包含概念 C_1;
- 实例检验($\Sigma \models C(i)$): 用来检验个体 i 是否属于概念 C。

2.2　面向 Z39.50 协议的 EHDM 的 DL 封装机制

EHDM 能够直接反映原始数据源模型的结构以及语义关系, 因此可以在 EHDM 上直接面向 Z39.50 进行 DL 封装, 其中 EHDM 的两类节点实体、关系实体以及约束可以分别表示为 DL 的基础概念和角色, 而 Z39.50 协议中的 APs 则直接定义为 EHDM 虚拟全局视图的节点实体。

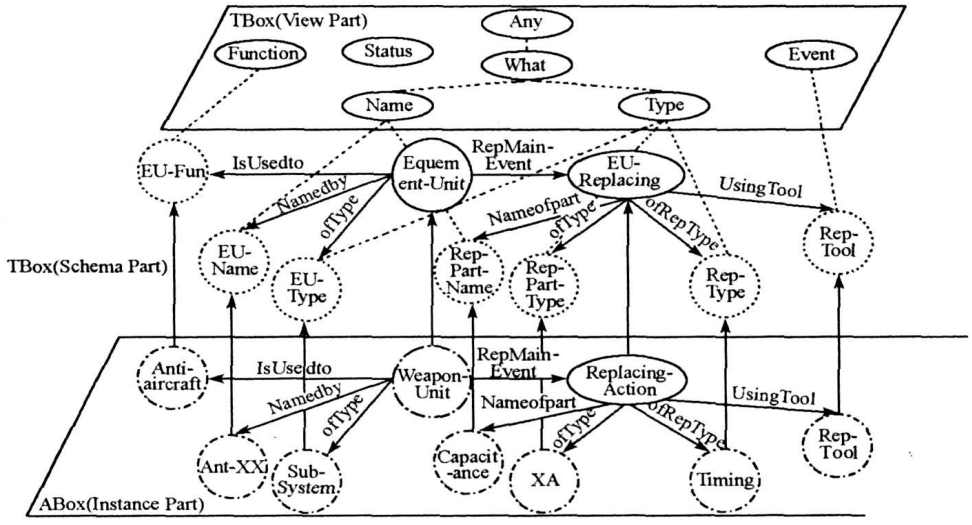

图 2　面向 Z39.50 协议的分层映射

Fig. 2　Z39.50 oriented layer mapping

如图 2 所示, 中间层表示模式部分, 用来直接表示数据源模型, 在 DL 中用基本概念表示, 上层为面向用户查询的导出模式, 表示为视图部分, 在 DL 中由导出概念表示, 两种视图都作为 Z39.50 协议中的 APs, 用 DL 中的 TBox 表示。图 2 下层表示 EHDM 模式实例, 在 DL 中由 ABox 表示。

3　EHDM 的 DL 封装在 Z39.50 协议中的应用

3.1　DL 概念与 EHDM 节点及 AP 映射关系及转换

(1) 直接映射: AP 直接对应 EHDM 模式部分的一个节点, 如图 2 中的 Function 对应 EHDM 的 EU-Fun, 表示为: $Function \bullet EU\text{-}Fun$。

(2) 组合概念映射: 大多情况下, AP 对应 EHDM 中的一些模式结构元素的运算结果集合, 以并集运算为例, 图 2 中 Name 对应 EU-Name 和 Rep-Part-Name 的并集, 表示为: $Name \bullet EU\text{-}Name \cup Rep\text{-}Part\text{-}Name$, Type 也是类似的情况, 运算可以为表 1 给出的任何一种, 比较常见的为 $\cup$ 和 $\cap$ 等集合运算。

(3) 对于为 Any 的 AP, 采用 $Any = Who \cup What \cup When \cup Where$ 机制来进行映射。

(4) 复杂的映射关系可以用表 1 给出的运算来实现。

举例说明: 图 2 中, 对于替换维修事件可以通过维修类型来限定维修工具, 从而映射到事件 Event 上, 表示为: $Event \bullet \nu(UsingTool)^{-1}.(\exists\, gfRepType.\{''Timing''\}))$, 表达式中 $''Timing''$ 表示 DL 系统中的个体, 对应 EHDM 的数据实例, $\exists\, gfRepType.\{''''\}$ 表示替换维修行为的维修类型属性, 整个表达式表示了具有维修类型属性值为 $''Timing''$ 的维修行为采用的工具集合。

(5) 对于某些 APs 在数据源中没有对应信息的情况, 则 DL 将其映射到表 1 中 $\underline{\perp}$ 或 $\bar{r}$ 上。

可以通过 ABox 的推理服务进行映射有效性检验, 该检验不用访问底层数据源, 减少了查询开销。

3.2 DL导出概念与Z39.50协议查询的映射与转换

DL既可以作为知识表示语言,也可作为查询语言,Z39.50查询可表示为DL的导出概念。查询可认为是个体在查询结果集中需要满足的充要条件;反之,基础或导出概念可以通过解释来进行查询。Z39.50查询可转换成DL基础概念的导出形式,例如查询:$Q1$: $Name = "Ant\text{-}XX"$, $Name$ 为 AP,映射为基础概念EU-Name和Rep-Part-Name的导出概念 C_{AP},Ant-XX映射为DL中的个体 i,首先DL推理服务对 $C_{AP}(i)$ 进行实例检验: $\Sigma \models C_{AP}(i)$,可以避免盲目的数据源信息的浏览,查询结果为EU-Name和Rep-Part-Name中包含Ant-XX的个体,因此可将DL系统个体定义为Z39.50的查询结果。

给定一个DL知识基础 Σ,一个AP导出概念 C_{AP} 和一个具有形如 $AP = a$ 的Z39.50查询 q,q 的结果集合由概念 C_q: $\Gamma(C_q) = \{a \in O_\Sigma \mid \Sigma \models C_{AP}(a)\}$ 的解释给出,其中 O_Σ 表示了 Σ 的ABox的个体集合。Z39.50协议的查询结果返回概念中心关联的个体集合,而直接的AP查询返回的是AP关联的个体,因此需要对DL查询进行重写,首先给出路径表达式的定义:

定义2 一个路径表达式 P_{AP} 是一个元素序列: $p = e_1, e_2, \cdots, e_{n-1}, e_n$,对于 $i \in [1, n-1]: e_i \in \{\exists\} \cup \{\forall\} \cup \mathcal{R}$,其中 $\mathcal{R}$ 为基础角色名称集合,后缀为"$\cdot$"且 $e_n \in C$,为基础概念集合。

路径表达式可以看成概念间联系的纽带,$Event \cdot \forall(UsingTool)^{-1}.(\exists\, ofRepType.\{"Timing"\}))$ 给出了一个包括UsingTool,Event以及Rep-Type的概念路径。复杂的查询需要多个概念域的概念的联合,通过路径表达式定义概念的组合来实现比较复杂的查询。下面给出DL查询的重写过程:(1)首先将Z39.50查询翻译成基础DL查询概念。(2)将得到的DL表达式进行展开,通过迭代代入包含AP导出概念的基本组成部分的方式来实现。(3)Z39.50的最终表达式通过路径表达式来实现,该路径表达式包括基本的概念。

通过DL推理的其他服务可进一步简化查询过程,例如根据 $\Sigma \models C_1 \prec C_2$ 可判断概念之间包含性,进而重复利用先前查询结果。

4 小结

EHDM具有较强的表达能力,通过添加动态行为可以增强语义表达能力,Z39.50协议通过APs支持异构数据源的虚拟集成和访问,本文对EHDM进行层次抽象,以EHDM的两类节点和边作为APs,在数据源庞大、动态性和自制性强,尤其是无法准确获取数据源语义的情况下更准确地执行信息检索,通过在EHDM和Z39.50之间建立DL层,将DL系统的概念、导出概念等分别映射到APs以及Z39.50查询上,可解决原有Z39.50协议查询对结构化数据源支持不好、空结果及错误结果频繁出现等问题,采用类DataSpace的三层结构能够实现根据需求集成的"量入为出"的数据集成策略,在海量数据环境下,相比集中式的集成方法具有更好的可行性,可减少集成的工作量,采用分层结构定义EHDM,可根据用户需求修改上层概念节点,而不需要破坏数据本身的完整性,因此具有较好的扩展性。

参考文献:

[1] Smith A C, Peter M. Inter Model Data Exchange of Type Information via a Common Type Hierarchy[C]//DISWeb06, Luxembourg, 2006: 307-321.

[2] Alon H, Michael F, David M. Principles of Dataspace Systems[C]//PODS'06, Chicage, Illinois, USA, 2006.

[3] Peter M, Alexandra P. Automatic Migration and Wrapping of Database Applications-a Schema Transformation Approach[C]//ER99, Springer Verlag LNCS, Paris, 1999: 96-113.

[4] Peter M, Alexandra P. Data Integration by Bi-Directional Schema Transformation Rules[C]//ICDE03, Boston, 2003: 227-238.

[5] Peter M, Alexandra P. Schema Evolution in Heterogeneous Database Architectures, a Schema Transformation Approach[C]//CAiSE02, Toronto Canada, 2002: 484-499.

[6] Noy N F. Semantic Integration—A Survey of Ontology-based Approaches[C]//SIGMOD Record, 2004, 33(4): 88-101.

[7] Yannis V, Vassilis C, Panos C. On Z39.50 Wrapping and Description Logics[J]. International Journal on Digital Libraries(JODL), 2000, 3(3): 208-220.