中文叙词表本体一致性检测机制研究与实现*

曾新红　林伟明　明仲

(深圳大学图书馆，深圳，518060) (深圳大学信息工程学院，深圳，518060)

［摘要］本文研究了中文叙词表本体（OntoThesaurus，即基于中文叙词表建立的本体知识库）的一致性检测机制，并将其应用在中文叙词表本体共建共享系统（OTCSS）的修订意见提交、叙词表本体更新和全局检查等相关过程的实现中，取得了良好的应用效果。

［关键词］叙词表　本体　中文叙词表本体　本体构造　一致性检测　本体演化

［分类号］G254　TP311

Research and Implementation of consistency checking mechanism for OntoThesaurus

Zeng Xinhong　Lin Weiming

(The Library of Shenzhen University, Shenzhen 518060, China)

Ming Zhong

（College of Information Engineering of Shenzhen University, Shenzhen 518060, China）

[Abstract] This paper studies the consistency checking mechanism for OntoThesaurus. It is used in OTCSS (OntoThesaurus Co_construction and Sharing System) to realize the submission of amend opinion, updating and global checking of OntoThesaurus, and works effectively.

[Keywords] thesaurus, ontology, OntoThesaurus, ontology construction, consistency checking, ontology evolution

1　前言

叙词表是本体出现之前最高端的知识组织系统，其编制有严格的国际和国家标准规范。但由于我国现有的主题词表（叙词表）大部分是手工编纂的，难免出现错误。在将其进行了本体化升级之后，可以借助本体语言的推理能力对其进行严格的一致性检测，理清和补全相关信息，从而建立起严格的概念体系和词间关系体系，极大地提高叙词表的科学性。

中文叙词表本体（OntoThesaurus）是基于中文叙词表建立的本体知识库。中文叙词表本体共建共享系统 OTCSS 将已有的中文叙词表转换为 OWL 文件，并通过在其网络共享应用过程中采集使用者知识来实现其动态更新完善。初始 OntoThesaurus 首先要进行一次全局一致性检测，以修正叙词表中原有的错误。随后，在 OntoThesaurus 的共建过程中，一致性检测机制的应用可以大大降低修订意见提交者和修订专家的工作量和错误率，并保证更新后的 OntoThesaurus 不出现新的一致性冲突。因此，一次性检测机制的研究和实现对 OntoThesaurus 在整个生命周期中的健康运行至关重要。

2　OntoThesaurus 的 TBOX 构建

OntoThesaurus 的类定义和属性定义请参见参考文献[1]中的表 1 和表 2。本文作了进一步的研究，进行了以下修改和扩展：

• 直接以叙词作为概念的表述形式，取消 *Term* 类、*PTerm* 类和 *HasPTerm* 属性。此举大大缩小了本体的容量并简化了实现过程。

• 参考 ANSI/NISO Z39.19-2005[2]的第 8 节（Relationships），对 *Broader*，*Narrower* 和

*本文系国家社科基金项目“基于本体和知识集成实现中文叙词表的升级、共享和动态完善”（项目编号 05CTQ001）和国家自然科学基金项目“视角理论及其在本体集成中的应用”（项目编号 60673122）的研究成果之一。敦煌研究院信息资料中心为本文提供了重要的研究实例——《敦煌学检索词表》，特在此向李鸿恩老师及其同仁表示衷心的感谢！

Related 三个属性分别进行了子属性扩展，详见参考文献[3]。此举利于将初始的粗粒度 OntoThesaurus 逐渐演化为细粒度本体，从而支持基于概念间具体子关系的推理。

3 OntoThesaurus 中存在的一致性问题

我们认为，用于人工智能目的和推理目的的本体应该是严格控制的知识组织系统，其规范程度不应低于叙词表。因此我们在 OntoThesaurus 中保留了叙词表结构中的精华（即须符合汉语叙词表编制规则（GB13190-91）[4]中的核心要求），使叙词表界几十年来在术语控制上的研究成果得以延续。同时 OntoThesaurus 也是一个本体，必须符合本体理论和技术的要求，这就要求我们必须将叙词表中供人理解的规则表述明确定义为机器可理解的形式化规范说明。本体技术的应用也允许我们在保持其功能的前提下抛弃叙词表规则中纯粹为方便人工使用而制定的一些规则（如非叙词款目可不再存在）。

OntoThesaurus 的一致性问题具体表述如下。

1) 叙词必须一词一义。[4]

叙词因其在标引和检索中的特定功能而要求绝对单义[4]。这是术语控制的重要一步。

隐含：一个概念在 OntoThesaurus 中只能由一个叙词来表示；叙词在 OntoThesaurus 中必须明确定义；入口词不能与叙词同形。

2) 等同关系，指叙词与非叙词之间的关系。[4]

我们在 OntoThesaurus 中取消了“等同关系必须是双方互相指引”的规则，即只保留叙词款目，取消非叙词款目。此举对降低系统的实现复杂度有重要意义。检索时同样可从入口词检索到叙词，非叙词的指引作用仍存在。输出书本式叙词表格式时也可由程序自动生成非叙词款目。

3) 属分关系，指上位叙词与下位叙词之间的关系，必须相互指引。[4]

隐含：属分关系是叙词之间的关系；属分关系是互逆的；属分关系是与直接上下位词的关系，不可越级，否则无法生成层次化的词族等级。

叙词款目中显示直接属分关系，有利于使用者通过词间关系明确词义，并可启发读者进行扩检和缩检，我们认为是合理的冗余，可为系统减少不必要的推理负担。因此在 OntoThesaurus 中仍规定属分关系必须成对出现。

扩展的属分关系子关系也必须遵守以上规则。

4) 相关关系，指叙词之间属分以外的相关关系。[4]

隐含：相关关系是叙词之间的关系；相关关系不能与属分关系（及其子关系）重合。

以上规则同样适用于扩展的相关关系子关系。

5）为了有效控制 OntoThesaurus 的规模，避免出现低级错误和不必要的冗余，我们还明确了以下隐含规则：

所有词间关系都是反自反的，即不能是术语与其自身之间的关系；除相关关系外，其他词间关系都是反对称的，即关系两边的概念不可互换。

上述规则在叙词表编制规则中虽然没有明确提出，但作为一种常识性的共识，在各种叙词表的实际编制过程中已作为一种默认的潜规则而得到严格执行。

4 OntoThesaurus 一致性问题的形式化描述

本节我们用形式化方法来明确定义以上一致性问题。

4.1 叙词定义缺失问题

在叙词表中，除了等同关系是叙词与非叙词之间的关系外，属分关系和相关关系都是叙词之间的关系，且叙词必须明确定义。相应的，在 OntoThesaurus 中，除了 *HasNTerm* 以外的其他 *ObjectProperty* 都是概念（*Concept*）之间的关系，而 *Concept* 的实例在 OntoThesaurus 中必须明确定义。其形式化定义为：设有关系 *R*，若存在 *x,y* 满足 *R(x,y)*，且 *R* 的值域为 *{x|Concept(x)}*, 而在 OntoThesaurus 中未明确定义 *Concept(y)*，则判定 *y* 缺失定义。

运用“未定义叙词”检测可以查出在叙词款目中作为属、分、参、族等关系词出现，而又未明确定义为叙词的术语。

4.2 值域不一致问题

值域是指属性的取值范围，其形式化定义为：设 *C* 为本体中的概念，*C* 有属性 *R*，则 *R* 的值域表示为：$range(R) = \{y|\exists x\ (R(x,y) \wedge C(x))\}$ 。

值域不一致是指知识框架中的属性取值不在定义的值域范围内。其形式化定义为：设 *C* 为本体 *M* 中的概念，*C* 有属性 *R*，如果 *M* 中存在 *R(c,c')*，其中 *C(x)* 且 $c' \notin range(R)$，则 *M* 存在值域不一致。

OntoThesaurus 中包含叙词 *Concept*、非叙词 *NTerm* 以及词间的代 *HasNTerm*、属 *Broader*、分 *Narrower*、族 *TopConcept*、参 *Related* 等关系。其中 *Broader*、*Narrower*、*Related*（及其子关系）和 *TopConcept* 等关系（属性）的值域均是*{x|Concept(x)}*（即它们都是叙词之间的关系），而代关系 *HasNTerm* 的值域是*{x|Nterm(x)}*，且 $Concept \cap NTerm = \varnothing$。因此 *OntoThesaurus* 可能存在值域不一致的问题。

运用“值域不一致”检测可以查出入口词与叙词同形的错误（即一个术语既是叙词又是入口词）。

4.3 OntoThesaurus 中的 *HasNTerm* 关系是反函数型的

反函数型定义如下：设 *R* 为定义在集合 *X* 上的二元关系，$\forall x \forall y \forall z\ ((R(x,z) \wedge R(y,z)) \rightarrow x=y)$，则称 *R* 是反函数型的。

在 OntoThesaurus 中规定，代关系属性 *HasNTerm* 是反函数型的，即一个入口词不能出现在多个叙词之下。形式化定义为：若关系 *R* 是反函数型的，存在 *x,y,z* 满足 $Concept(x) \wedge Concept(y) \wedge Concept(z) \wedge R(x,z) \wedge R(y,z) \wedge x \neq y$，则 OntoThesaurus 出现一致性问题。

本条规则通过检测入口词的多次出现，可检测出同一个概念在 OntoThesaurus 中出现多个叙词的情况。若确系不同概念使用了同一个入口词，则允许例外（或为入口词添加限定词进行区分）。

4.4 OntoThesaurus 中的所有词间关系是反自反的

为了控制 OntoThesaurus 的规模，保持较小的冗余度，规定其中的所有词间关系（*HasNTerm*、*Broader*、*Narrower*、*TopConcept*、*Related* 以及所有扩展的子关系）均应是反自反的（即不能是术语与其自身之间的关系）。

反自反的定义是：设 *R* 为定义在集合 *X* 上的二元关系，如果 $\forall x \in X\ (x,x) \notin R$，则称 *R* 是反自反的。如果 OntoThesaurus 中的词间关系 *R*，存在 *x* 满足 *Concept(x)*且 *R(x,x)*，那么判定 OntoThesaurus 是不一致的。

4.5 除相关关系（*Related*）外，OntoThesaurus 中的其他词间关系都是反对称的

反对称的定义是：设 R 为定义在集合 X 上的二元关系，如果 $\forall x \forall y\ ((R(x,y) \wedge R(y,x)) \rightarrow (x=y))$，则称 R 是反对称的。对于 OntoThesaurus 中 *Related* 之外的关系 R，如果存在 x,y 满足 $Concept(x),Concept(y)$且有 $x \neq y, R(x,y), R(y,x)$，则判定 OntoThesaurus 是不一致的。

运用“非法对称关系”检测可检查出某些低级错误，例如 A 属 B 而 B 又属 A 的情况。

4.6 OntoThesaurus 中任意两个词间关系（*TopConcept* 除外）之间不能存在同一个二元组

除了 *TopConcept* 与 *Broader* 关系可能共享同一个二元组（即一个叙词款目中的属关系词和族首词是同一个叙词）外，OntoThesaurus 的所有词间关系的任意两个关系之间不能存在同一个二元组。形式化定义为：设 R_1,R_2 是 OntoThesaurus 中的词间关系，且 $R_1 \neq R_2$，如果存在 x,y 满足 $Concept(x) \wedge Concept(y) \wedge R_1(x,y) \wedge R_2(x,y)$，那么 OntoThesaurus 是不一致的。

运用“二元关系冲突”检测可检查出某些词间关系错误，例如 A 分 B 同时 A 又参 B 的情况。

4.7 OntoThesaurus 中互逆的一对关系的断言必须成对出现

逆关系的定义是：设 R 为 X 到 Y 的二元关系，R 的逆关系 $R^{-1}=\{(y,x)|R(x,y)\}$。

在 OntoThesaurus 中，互逆的一对关系（如 *Broader* 和 *Narrower*）的断言必须成对出现。也就是说，设 R_1,R_2 为 OntoThesaurus 中的两个关系，且$(R_1)^{-1}= R_2$，存在 x,y 满足 $Concept(x) \wedge Concept(y) \wedge R_1(x,y)$，如果$(y,x) \notin R_2$，那么判定 OntoThesaurus 出现信息缺失，须补充缺失的信息。

4.8 OntoThesaurus 中具有传递性的关系的断言不能出现越级

OntoThesaurus 中的属分关系及其子关系是具有传递性的。传递性的定义是：设 R 为定义在集合 X 上的二元关系，如果 $\forall x \forall y \forall z\ ((R(x,y) \wedge R(y,z)) \rightarrow R(x,z))$，则称 R 是传递的。而参照叙词表编制标准的规定，在 OntoThesaurus 中这些具有传递性的关系，它们的断言只反映当前叙词上下一级的属分关系，不能出现越级情况，这样才能保证能够根据属分关系推理出严格的词族等级结构。其形式化定义是：在 OntoThesaurus 中，设 R 具有传递性，定义关系 R'如下：

$$\forall x \forall y\ (R(x,y) \rightarrow R'(x,y))$$

$$\forall x \forall y \forall z\ ((R'(x,y) \wedge R'(y,z)) \rightarrow R'(x,z)).$$

如果存在 x,y,z 满足 $Concept(x) \wedge Concept(y) \wedge Concept(z) \wedge R(x,y) \wedge R(x,z) \wedge R'(y,z)$，则判定 OntoThesaurus 存在越级情况，是不一致的。

系统可以根据通过了一致性检测的属分关系推理出严格的词族等级结构，并自动补充族首词。

5 OntoThesaurus 一致性检测机制的实现

我们运用 Jena 提供的基于自定义规则的推理机实现了 OntoThesaurus 的一致性检测机制。

Jena 是 HP 公司的开源语义网应用框架，它为 RDF、RDFS 和 OWL 提供了可编程环境。Jena 的推理机制有三种：使用 Jena 自带的基于一般规则的推理机、使用自定义规则的推理机、使用外部推理机。Jena 开发包对 OWL 的推理提供了完备的支持[5]。

5.1 自定义规则

Jena 提供基于自定义规则的推理机[5]，它通过一定的推理引擎来解释这些规则，并完成推理。用户可根据需要定制自己的规则，然后创建特定的推理机来完成推理。Jena 的推理机提供前向链、后向链和混合式的推理引擎。其中，前向链和后向链推理引擎可以独立使用，也可以使用前向链引导后向链推理引擎。本文采用前向链推理引擎来实现 OntoThesaurus 的一致性检测机制。前向链推理引擎基于标准的 RETE 模式匹配算法，该算法由 Charles Forgy 博士在 1979 年提出，是在模式匹配中利用推理机的时间冗余性和规则结构的相似性，通过保存中间运算来提高推理效率的一种模式匹配算法。算法的核心思想是对分离的匹配项根据内容来动态构造匹配树，以达到降低运算量的效果[6][7]。

5.2 运用自定义规则检查一致性

在上两节中讨论的 OntoThesaurus 的一致性问题可以通过使用 Jena 的自定义规则推理机来解决。具体的自定义规则如下：

1） 叙词定义缺失问题。解决思路如下：先运用规则*[r2: (?x rdfs:subClassOf ?y)(?a rdf:type ?x) -> (?a rdf:type ?y)]*来补全子类的实例，然后运用规则*[r1: (?p rdfs:range pre:Concept) (?x ?p ?y) (?x rdf:type pre:Concept) noValue(?y rdf:type pre:Concept) -> (?y rdf:type pre:Concept) (?y pre:err 'error1')]*来查出缺失定义的叙词。

2） 值域不一致问题。解决思路如下：

先运用规则*[r2: (?x rdfs:subClassOf ?y)(?a rdf:type ?x) -> (?a rdf:type ?y)]*来补全子类的实例，然后运用规则 *[r12: (?p rdfs:range pre:NTerm) (?x ?p ?y) (?y rdf:type pre:Concept) -> (?y pre:err 'error2')]*来查出值域不一致的问题。

3） OntoThesaurus 中的 *HasNTerm* 关系是反函数型的。解决思路如下：

运用规则 *[r10: (?x pre:HasNTerm ?y) (?ox pre:HasNTerm ?y) notEqual(?x,?ox) makeTemp(?z) -> (?z pre:err 'error10')]* 来查出不一致的地方。

4） OntoThesaurus 中的所有词间关系均是反自反的。解决思路如下：

先运用规则*[r2: (?x rdfs:subClassOf ?y)(?a rdf:type ?x) -> (?a rdf:type ?y)]*来补全子类的实例，然后运用规则*[r5: (?p rdf:type owl:ObjectProperty) notEqual(?p,pre:TopConcept) (?x ?p ?y) (?x rdf:type pre:Concept) equal(?x,?y) makeTemp(?z) -> (?z pre:err 'error5')]*来查出不一致的地方。注意，这里不考虑族首词这个角色，在一致性检查后每个实例的族首词可通过程序自动生成。

5） 除 *Related* 外，OntoThesaurus 中的其他词间关系都是反对称的。解决思路如下：

先运用规则*[r2: (?x rdfs:subClassOf ?y)(?a rdf:type ?x) -> (?a rdf:type ?y)]*来补全子类的实例，然后运用规则*[r4: (?p rdf:type owl:ObjectProperty) notEqual(?p,pre:Related) (?x ?p ?y) (?y ?p ?x) notEqual(?x,?y) (?x rdf:type pre:Concept) makeTemp(?z) -> (?z pre:err 'error4')]* 来查出不一致的地方。

6） OntoThesaurus 中任意两个词间关系（*TopConcept* 除外）之间不能存在同一个二元组。解决思路如下：

运用规则*[r6: (?x ?p ?y) (?p rdf:type owl:ObjectProperty) notEqual(?p,pre:TopConcept) (?x ?q ?y) notEqual(?q,pre:TopConcept) notEqual(?p,?q) (?q rdf:type owl:ObjectProperty) (?x rdf:type pre:Concept) makeTemp(?z) -> (?z pre:err 'error6')]* 来查出不一致的地方。（注意，这里不考虑族首词这个角色，原因同 4）。

7） OntoThesaurus 中互逆的一对关系的断言必须成对出现。解决思路如下：

运用规则*[r8: (?p owl:inverseOf ?q) (?x ?p ?y) noValue(?y ?q ?x) (?x rdf:type pre:Concept) (?y rdf:type pre:Concept) makeTemp(?z) -> (?z pre:err 'error8')]*来查出信息缺失的地方。

8） OntoThesaurus 中具有传递性的关系的断言不能出现越级。解决思路如下：

由于属分关系是具有传递性的，且它们互为逆关系，因此在补全了逆关系的情况下只需检查其中一种情况即可。建立一临时的角色 *TBroader*（代表任意级的上位关系），先运用以下三条规则*[r9a: (?a ?p ?b) (?p rdfs:subPropertyOf pre:Broader) (?a rdf:type pre:Concept) (?b rdf:type pre:Concept) -> (?a pre:Broader ?b) (?a pre:TBroader ?b)]*， *[r9b: (?a pre:TBroader ?b) (?b pre:TBroader ?c) (?a rdf:type pre:Concept) (?b rdf:type pre:Concept) (?c rdf:type pre:Concept) -> (?a pre:TBroader ?c)]*和 *[r9c: (?a pre:Broader ?b) (?a rdf:type pre:Concept) (?b rdf:type pre:Concept) -> (?a pre:TBroader ?b)]*找出所有传递级别的三元组，然后运用规则*[r9d: (?a pre:Broader ?b) (?a pre:Broader ?c) notEqual(?b,?c) (?b pre:TBroader ?c) (?a rdf:type pre:Concept) (?b rdf:type pre:Concept) (?c rdf:type pre:Concept) makeTemp(?z) -> (?z pre:err 'error9')]*来查出越级情况。

8) 此外，还可以利用自定义规则推理为 OntoThesaurus 自动补全族首词。解决思路如下：

运用以下规则*[r13a: (?a ?p ?b) (?p rdfs:subPropertyOf pre:Broader) -> (?a pre:Broader ?b)][r13b: (?a pre:Broader ?b) (?b pre:Broader ?c) -> (?a pre:Broader ?c)][r13c: (?x rdfs:subClassOf ?y) (?a rdf:type ?x) -> (?a rdf:type ?y)]*来补全叙词之间的上位关系（包括间接的上位关系），然后再运用规则*[r13c1: (?a pre:Broader ?b) noValue(?b pre:Broader ?c) -> (?a pre:TopConcept ?b)]*来自动生成各个叙词的族首词。

5.3 OntoThesaurus 一致性检测机制的具体实现

我们运用 Jena 开发包来具体实现 OntoThesaurus 的一致性检查机制。步骤如下：

1） 运用 Jena 提供的 ModelFactory 来创建一个 Ontology Model，调用该 Model 的 read 方法将 OWL 格式的 OntoThesaurus 读入 Ontology Model 中。

2） 根据具体要求写出对应的推理规则。

3） 运用 Jena 提供的 GenericRuleReasonerFactory 读入推理规则形成推理器 Reasoner。

4） 根据 Reasoner 以及第三步创建的 Ontology Model 来创建推理模型 InfModel。

5） 可调用 InfModel 的 prepare 方法来触发推理规则的运行，也可通过读取推理模型的信息来触发。

6 OntoThesaurus 的一致性检测机制的应用效果

目前，OntoThesaurus 的一致性检测机制已应用在中文叙词表本体共建共享系统（OTCSS）的修订意见提交、叙词表本体更新和全局检查等相关过程的实现中，取得了良好的效果。下面以《敦煌学检索词表本体共建共享系统》为例，着重介绍分步全局检查的步骤和应用效果。在修订意见提交、叙词表本体更新过程中，系统有针对性地运用了以下这些步骤的局部或全部，以减少修订意见提交者和修订专家的工作量和降低他们的出错率，并保证 OntoThesaurus 在整个生命周期中的健康运行。

对于刚刚从已有的中文叙词表转换而来的初始 OntoThesaurus 来说，由于大部分现有中文叙词表是手工编制的，可能存在比较多的一致性问题，可以先按以下顺序逐步检测和排除一致性问题，最后再进行一次批式的全局检查。按以下顺序检测可以使系统和修订专家的工作量最小化，因为前一个问题可能是导致后一个问题出现的重要因素。而对于直接使用

OTCSS 系统建立（即将已有叙词表输入或完全新建）的 OntoThesaurus 来说，由于在建设过程中已经过了比较严格的一致性检测，发生错误的可能性较小，可以在发布共享之前直接进行一次批式的全局检查，以从全局的角度发现和排除最后的错误。

6.1 未定义叙词

“未定义叙词”检测可查出在已有叙词款目的词间关系中出现而又未定义为叙词（即不存在叙词款目）的术语。修订专家可通过网络界面查看相关款目详细信息，判断是否应为其添加叙词款目，然后通过自动添加或修改/删除已有款目相关信息等手段进行处理。

未定义叙词的检测使 OTCSS 系统可以在叙词款目的保存过程中自动添加相关叙词的叙词款目（带有自动生成的拼音和由互逆关系推出的词间关系等信息），从而可大大降低输入工作量，加快建库过程。

6.2 值域不一致的问题

通过值域不一致检测可查出既是叙词又是入口词的术语（见图 1）。例如，如图 2 所示：“维摩经疏”本身是一个叙词，同时它又是另一个叙词“维摩经义疏”的入口词。修订专家可通过网络界面进行修改处理，消除冲突。

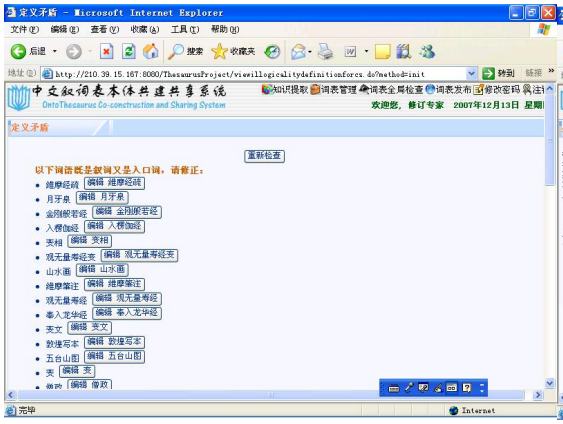

图 1 值域不一致检测结果

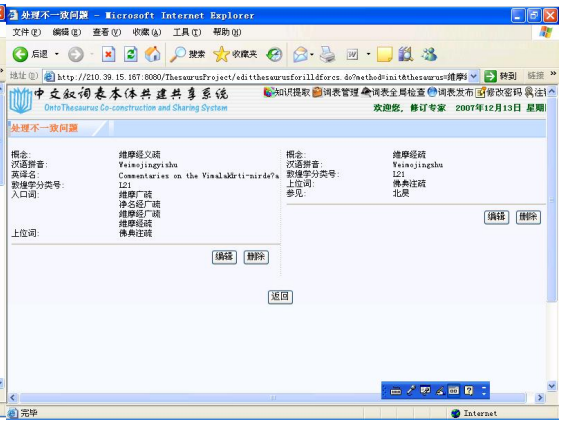

图 2 值域不一致示例

6.3 入口词多次出现

通过“入口词多次出现”检测可以检查出以下错误：为同一概念选用了多个叙词，或由于输入错误未能查重而造成同一概念出现多个叙词。此时应只保留一个叙词，其他作为入口词出现或予以删除（如图 3 图 4 所示）。

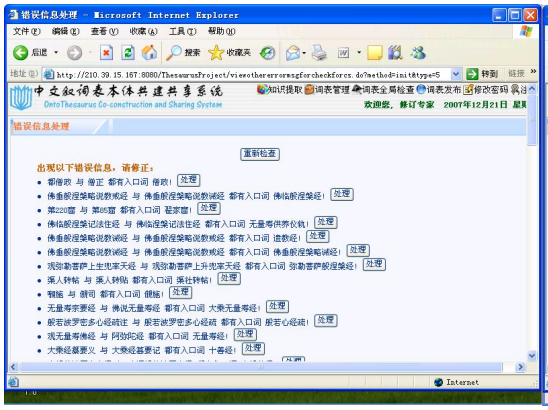

图 3 入口词多次出现检测结果

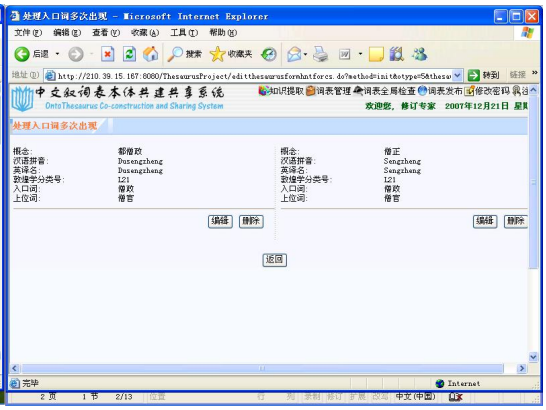

图 4 入口词多次出现示例

还有一种特殊情况就是，不同的概念（叙词）确实存在同形的入口词。例如在《敦煌学检索词表》中，古“空谷山”、“三危山”和“无穷山”今均俗称为“火焰山”，敦煌石窟代表窟“第 220 窟”和“第 85 窟”亦均称为“翟家窟”，经查阅《敦煌学大辞典》[8]无误，建议保留。

6.4 非法自反关系

从逻辑角度来看，自反关系（即术语与其自身之间的关系）是关系的一个特例，在某种意义上不能算是一种错误。但自反现象的存在会徒然增加系统的实现和维护成本，因此在结构严谨的知识组织系统中有必要杜绝这种现象的存在。在传统的中文叙词表的编制过程中其实已默认执行了这条规则，所以这种现象只是偶有发生（《敦煌学检索词表》中仅检出 3 条）。在 OntoThesaurus 的动态完善过程中，非法自反关系的检测可有效控制 OntoThesaurus 的规模膨胀。

6.5 非法对称关系

通过非法对称关系检测可检查出手工编制词表中可能出现的某些低级错误，如两个术语都是叙词而又互为入口词的情况（详见图 5 图 6），或一个叙词与另一个叙词互为上位词等错误。

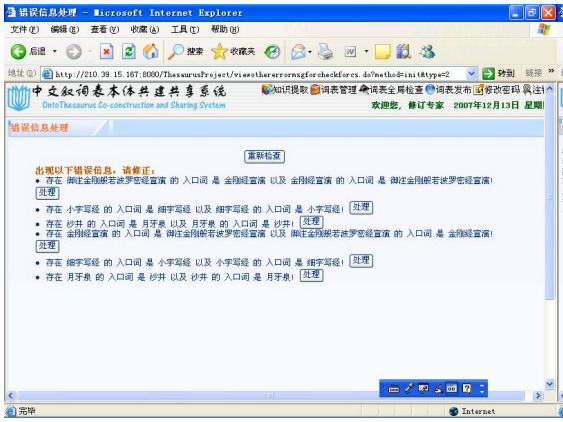

图 5 非法对称关系检测结果

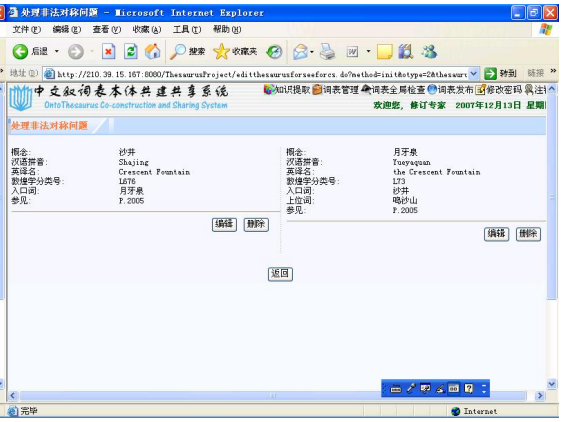

图 6 非法对称关系示例

6.6 未成对指引关系

未成对指引关系的检测可检查出必须成对出现的互逆关系的缺失，系统可做自动补齐处理或提示人工处理（详见图 7 和图 8）。属分关系未能相互指引是手工编制叙词表中的常见错误，通过这一步检测和修订处理可全部排除。通过 OTCSS 系统对 OntoThesaurus 进行动态完善的过程中，因有严格的未成对指引关系检测，可以自动补充添加成对指引关系，从而可杜绝此类错误的再发生。

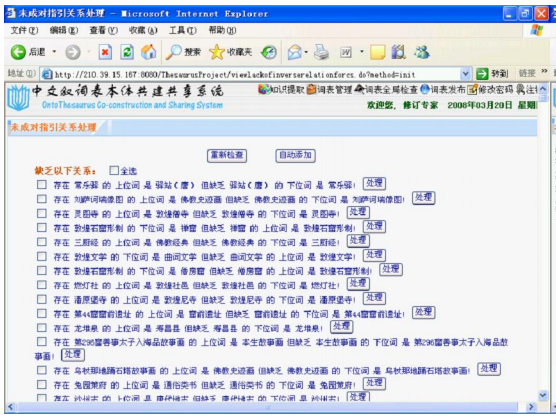

图 7 未成对指引关系检测结果

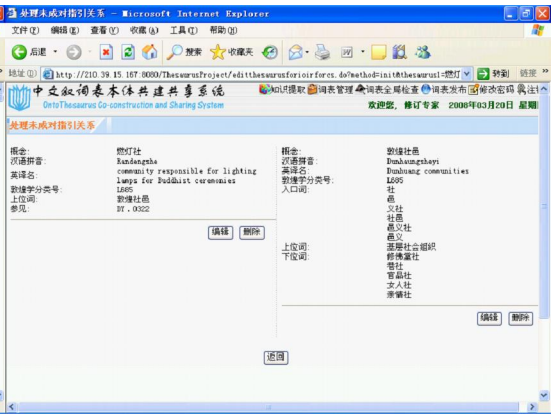

图 8 未成对指引关系示例

6.7 二元关系冲突

在手工编制的叙词表中，二元关系冲突偶有发生。例如，如图 9 和图 10 所示，在《敦煌学检索词表》中出现的一个二元关系冲突错误：“敦煌俗文学”**参**“说唱故事”同时又**分**“说唱故事”。此时修订专家只能选择保留其中一个关系。

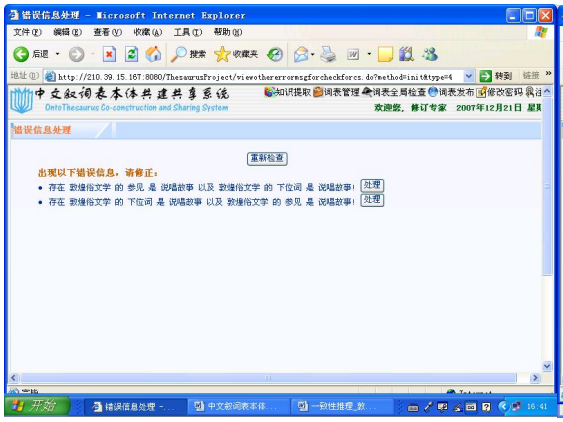

图 9 二元关系冲突检测结果

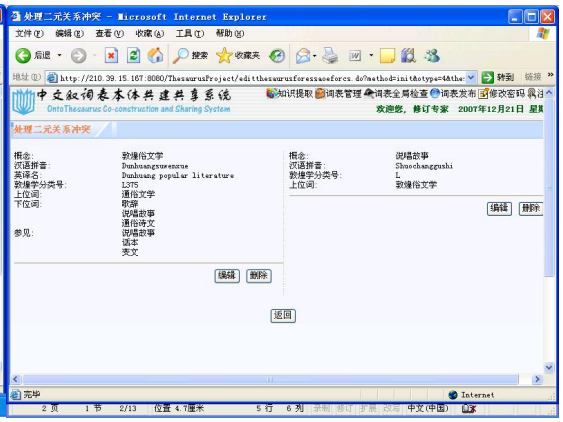

图 10 二元关系冲突示例

6.8 传递关系越级

为了获得层次分明的词族等级结构，在 OntoThesaurus 的叙词款目中只能出现最近一级的属分关系（或其子关系）。也就是说，传递关系不可越级。这是手工编制词表较难把握的问题，比较容易出现错误。通过传递关系越级检测可以找出并排除这种错误。例如，如图 11 和图 12 所示，在《敦煌学检索词表》中检测到的一个传递关系越级错误：“大辟图”同时**属**“经变画”和“敦煌壁画”，而“经变画”又**属**“敦煌壁画”。此时修订专家可通过网络界面比较分析此错误所涉及的三条款目的详细信息，在“大辟图”款目中删除越级的上位词“敦煌壁画”，即可理顺传递关系。

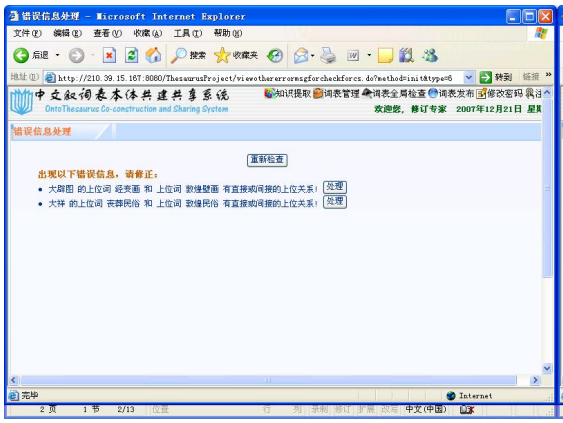

图 11 传递关系越级检测结果

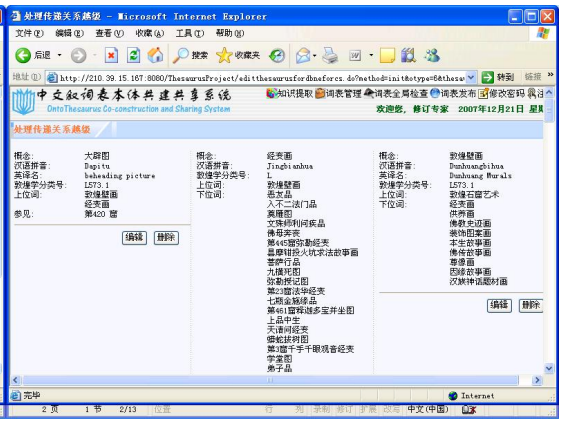

图 12 传递关系越级示例

6.9 自动生成族关系

在传统的叙词表修订过程中，需要在后期根据修订过的属分关系人工调整词族等级和族首词。而在 OntoThesaurus 中，这一步可以由机器自动完成。鉴于使用者调用查看整个词族并非高频率事件这个事实，我们选择采用使用者需要时再通过推理动态生成词族的方式提供服务（《中国分类主题词表》二版电子版也采用这种方式）。而族首词的存在有利于这一服务的实现，因此我们在全局检查的最后一步为具有属分关系的叙词自动生成族关系（族首词）。

若要求严格按属分关系的各种子关系来生成词族等级，则需要考虑为族关系增加相应的子关系（如类属族首词、实例族首词和整体/部分族首词）。

7 结 语

OntoThesaurus 一致性检测机制的实现有力地保证了中文叙词表本体在其生命周期每个阶段的质量。该机制对其他知识组织系统（如分类法、规范档等）和知识组织系统表示方法（如 SKOS[9]）具有较好的可移植性，对实现这些知识组织系统和表示方法的一致性检测具有直接的参考价值。

参考文献

[1] 曾新红.《中国分类主题词表》的 OWL 表示及其语义深层揭示研究[J]. 情报学报, 2005(2):151-160

[2] ANSI/NISO Z39.19-2005. Guidelines for the Construction, Format, and Management of Monolingual Controlled Vocabularies. Developed by National Information Standards Organization, approved July 25, 2005 by the American National Standards Institute

[3] 曾新红等.中文叙词表本体共建共享系统研究[J]. 情报学报，2008（3）

[4] 中华人民共和国国家标准, GB13190 —91 ,汉语叙词表编制规则. 中国标准出版社,1991.

[5] Jena-A Semantic Web Framework for Java[EB/OL]. http://jena.sourceforge.net/(Accessed Jul. 7, 2006)

[6] Forgy C L. Rete: A Fast Algorithm for the Many Pattern/Many Object Pattern Match Problem[J]. Artificial Intelligence. 1982, 19(1): 17-37

[7] The RETE Algorithm[EB/OL]. http://www.cis.temple.edu/~ingargio/cis587/readings/rete.html#2(Accessed Jul. 10, 2006)

[8] 季羡林主编. 敦煌学大辞典[M]. 上海辞书出版社，1998

[9] SKOS Use Cases and Requirements: W3C Working Draft 16 May 2007[EB/OL]. http://www.w3.org/TR/2007/WD-skos-ucr-20070516/(Accessed Jun. 8, 2007)