

一个使用Protege 5.5和插件版本3.2来构建

OWL本体的实用指南

8 2021年10月迈

克尔· 德贝利斯

这是由马修· 霍里奇编写的Protege 4教程版本1.3的修订版本。本教程的之前版本是由霍尔格· 克努布劳奇、艾伦· 雷克特、罗伯特· 史蒂文斯、克里斯· 沃罗、西蒙· 朱普、乔治娜· 莫尔顿、尼克· 德拉蒙德和塞巴斯蒂安· 勃兰特开发的。

这项工作使用了Protege资源进行，该资源得到了美国国立卫生研究院国家普通医学科学研究所GM10331601的资助。

第3-5章是基于最初的教程。我已经更新了这个教程，以与Protege 5保持一致。我还做了一些更改，以解决我看到的新用户遇到的一些最常见的问题，删除了一些关于旧的基于框架的Protege版本的过时信息，以及各种各样的更改。第6-11章是新的。我已经为SWRL、SPARQL和SHACL等技术添加了新的部分，以及一些关于iri和名称空间等概念的细节。

感谢马修· 霍里奇和所有参与之前教程的人。特别感谢洛伦兹· 布赫曼，他在我开发修改后的例子时帮助我解决了一个棘手的问题，也感谢安德烈· 沃尔斯基对SHACL插件的帮助。特别感谢迪克· 奥姆斯、科林· 皮尔金顿和利维亚· 皮内拉对本教程的之前版本提供了出色的详细反馈。特别感谢简· 阿斯曼，燕旭，和弗朗茨公司的每个人的帮助利用寓言图和格拉本教程和一个新的教程可用的at：https://www.michaeldebellis.com/post/the-people_example-本体也，感谢Protege用户支持电子邮件列表上的每个人。

注意：此文档可能会经常得到更新。查看一下我的博客是个好主意：

<https://www.michaeldebellis.com/post/new-protege-pizza-tutorial> to要确保你有最新的版本。

如果你有任何问题或评论，请随时联系我，atmdebellissf@gmail.com

内容

第一章简介	4
1.1 许可	4
1.2 惯例	4
第二章要求和门控仪用户界面	6
第三章OWL的本体论是什么？	6
3.1 OWL本体的组件	6
3.1.1 个人	7
3.1.2 属性	8
3.1.3 类	8
第四章建立OWL本体论	10
4.1 命名的类	13
4.2 使用推理器	15
4.4 使用创建类层次结构	17
4.5 创建Pizzatock层次结构	19
4.6 OWL属性	22
4.7 逆属性	23
4.8 OWL对象属性特征	24
4.8.1 功能属性	24
4.8.2 反函数性质	25
4.8.3 瞬态特性	25
4.8.4 对称的和不对称的性质	25
4.8.5 反射特性和不反射特性	26
4.8.6 推理器会自动强制执行属性特征	26
4.9 OWL属性域和范围	26
4.10 描述和定义类	29
4.10.1 财产限制	29
4.10.2 存在性限制	31
4.10.3 创建披萨的子类	33
4.10.4 检测不能具有成员的类	37
4.11 原始类和定义类（必要类和充分公理）	38
4.12 普遍限制	41
4.13 自动分类和开放世界推理	42

4.14 定义枚举类	44
4.15 添加香料作为一个属性	45
4.16 基数限制	46
第五章数据类型属性	48
5.1 定义数据属性	48
5.2 定制原型机用户界面	50
第6章为已枚举的类添加顺序	58
第七章名称：IRI 的名称、标签和名称空间	60
第八章：一个关于一些个体的更大的本体论	62
8.1 熟悉这个更大的本体论	63
第九章查询：描述逻辑和SPARQL	66
9.1 描述逻辑查询	66
9.2 SPARQL 查询	67
9.21 一些SPARQL披萨查询	67
9.22 SPARQL和IRI 名称	70
第十章SWRL和SQWRL	72
第11章SHACL	76
11.1 OWA和单调推理	76
11.2 现实世界是混乱的	76
11.3 基本SHACL概念	77
11.4 Protege SHACL插件	77
第十二章网络抗议	83
第十三章结论：一些个人的想法和意见	88
第十四章书目	89
14.1 W3C文件	89
14.2 网站、工具和演示文稿	89
14.3 论文	89
14.4 书	90
14.5 供应商	90

第一章简介

这介绍了用于创建OWL本体以及各种插件。如果您有本教程的问题，请随时发送电子邮件给我 `directly:mdebellissf@gmail.com`，但是，如果您有关于宝门，OWL或插件的一般问题，您应该订阅并发送电子邮件到用户支持宝门和网络电子邮件列表。这个列表中有很多人（包括我）来监控它，他们可以贡献他们的知识来帮助你理解如何最大限度地利用这项技术。要订阅这个列表，去 `to:https://protege.stanford.edu/support.php`and，点击第一个橙色的订阅按钮。这将使你能够订阅这个列表，并给你的电子邮件发送问题。

本章介绍了许可问题，并描述了本教程中使用的约定。第2章介绍了本教程的要求，并描述了Protege的用户界面。第三章简要概述了OWL的本体语言。第4章的重点是构建一个具有类和对象属性的OWL本体。第四章还描述了使用描述逻辑推理器来检查本体的一致性，并自动计算本体类的层次结构。

第5章描述了数据属性。第6章描述了设计模式，并展示了一种设计模式：向已枚举的类添加顺序。第7章描述了与OWL实体名称相关的各种概念。

第8章介绍了在第1-7章中开发的披萨教程的扩展版本。这个本体有少量已经创建的实例和属性值，这些值可以用于说明后面章节中用于编写规则、执行查询和定义约束的工具。

第9章描述了用于执行查询的两个工具：描述逻辑查询和SPARQL查询。第10章介绍了语义Web规则语言（SWRL），并指导您完成如何创建SWRL和SQWRL规则。第11章介绍了形状约束语言（SHACL），并讨论了在描述逻辑中定义逻辑公理与在SHACL中定义数据完整性约束之间的区别。第12章有一些总结性的思想和意见，第13章提供了一个参考书目。

1.1 许可

本文档可在知识共享属性-sharea类4.0国际公共许可下免费获得。我通常以PDF格式发布，但如果你想制作自己的版本，给我发邮件，我会给你Word版本。有关许可的详细信息，请参见：
<https://creativecommons.org/licenses/by-sa/4.0/legalcode>

1.2 惯例

类、属性、规则和个人名称都是用这样的统一字体写的。Protege和本文档中用于任何此类构造的术语是实体。个体和类也可以被称为对象。

用户界面选项卡、视图、菜单选择、按钮和文本条目的名称将如下高亮显示。

任何时候你看到高亮显示的文本，如文件>首选项或确定或点击，它指的是你应该或可以查看或输入用户界面的东西。如果您曾经不知道如何完成一些任务，请寻找突出显示的文本。通常，与顶点一样，您在Protege UI中输入的文本将是类的名称、属性等。在这些情况下

名称意味着输入一个字段，它将只高亮显示，而不是高亮显示并以统一字体打印。

菜单选项显示了顶级菜单的名称，然后是一个>，然后是下一级向下到所需的选择。例如，要指示如何在窗口菜单中的选项卡部分下打开按类的个人选项卡，将使用以下文本：窗口>选项卡>按类的个人。

当一个单词或短语被强调时，它以斜体显示。

练习介绍如下：

练习1：完成此操作

-
1. 这样做。
 2. 然后这样做。
 3. 然后这样做。
-



潜在的陷阱和警告都是这样呈现的。



关于使用蛋白质的提示和建议如下。



关于事物含义的解释是这样呈现的。



一般的说明是这样呈现的。



词汇解释和替代名称都是这样呈现的。

第二章要求和门控仪用户界面

为了遵循本教程，你必须要有在Protege网站上提供的Protege5.1和一些项目插件，这将在下面更详细地描述。现在，只要确保你有最新版本的机器人。在编写这篇文章的时候，最新版本是5.5，尽管该教程也适用于以后的版本。

Protege的用户界面被分为一组主要的选项卡。这些选项卡可以在Windows>选项卡选项中看到。此选项将显示当前已加载到Protege环境中的所有UI选项卡。当前打开的任何选项卡旁边都有一个复选标记。要看到一个不可见的选项卡，只要从菜单中选择它，它将被添加到顶部与其他主要选项卡，它的菜单项现在将被检查。您可以通过加载插件向环境中添加额外的主要选项卡。例如，当我们加载SHACL4Protege插件时，SHACLEditor将被添加到菜单中。

每个主要的选项卡由各种窗格或作为Protege称为他们的视图。可以使用每个视图的右上角的图标来调整每个视图的大小或关闭。这些视图也可以作为每个主要选项卡中的子选项卡进行嵌套。当选项卡是自己的屏幕（在窗口>选项卡选项下）和子选项卡视图之间可能存在混淆时，我们将称屏幕选项卡为主要选项卡。有许多视图不在默认版本的Protege中，可以通过Window>视图选项来添加。附加的视图被分为各种类别，如Window>Views>个人视图。第5.2节将显示一个向主选项卡添加新视图的示例。

第三章OWL的本体论是什么？

本体论用于捕获有关某些感兴趣的领域的知识。本体描述了领域中的概念，以及这些概念之间的关系。不同的本体语言提供了不同的设施。标准本体语言的最新开发是来自万维网联盟（W3C）的OWL。关于OWL的基本概念，有一个很好的引物是at：<https://www.w3.org/TR/owl2-primer/>

OWL使得基于集合理论和逻辑以明确的方式描述概念成为可能。复杂的概念可以由更简单的概念组成。逻辑模型允许使用一个推理器，该推理器可以检查本体中的所有语句和定义是否相互一致，还可以识别哪些概念符合哪些定义。因此，推理器可以帮助正确地维护层次结构。这在处理类可能有多个父类的情况时特别有用。推理者还可以推断出其他信息。例如，如果两个属性是逆的，那么用户只需要断言一个值，并且推理器将自动推断出该逆值。

3.1 OWL本体的组件

OWL本体由类、属性和个体组成。OWL本体是描述逻辑（DL）的一个实现，它是一阶逻辑的一个可供决定的子集。OWL中的一个类是一个集合，一个属性是一个二进制关系，而一个个体是一个集合中的一个元素。来自集合理论的其他概念也在OWL中实现，如不相交集，空集合（猫头鹰：没有），逆

¹ <http://protege.stanford.edu>

关系，传递性关系，等等。理解集合理论的基本概念将有助于用户最大限度地利用OWL，但这不是必需的。Protege的好处之一是，它提供了一个直观的GUI，使领域专家能够在没有集合理论背景的情况下定义模型。然而，开发人员被鼓励更新他们关于逻辑和集合理论的知识。一个很好的资料来源是刘易斯和帕帕达米里斯的《计算理论的要素》的前三章。另一个很好的资料来源是PDF文档的集合理论概述，可在：

<https://www.michaeldebellis.com/post/owl-theoretical-basics>

3.1.1 个人

个人代表了人们所感兴趣的领域中的对象。OWL与大多数编程和知识表示语言之间的一个重要区别是，OWL不使用唯一名称假设（UNA）。这意味着两个不同的名字实际上可以指同一个人。例如，“伊丽莎白女王”、“女王”和“伊丽莎白·温莎”可能都指的是同一个人。在OWL中，必须明确地指出个体彼此之间是相同的，或者彼此之间是不同的。图3.1显示了在人、国家和关系领域的一些个人——在本教程中，我们将个人表示为钻石。

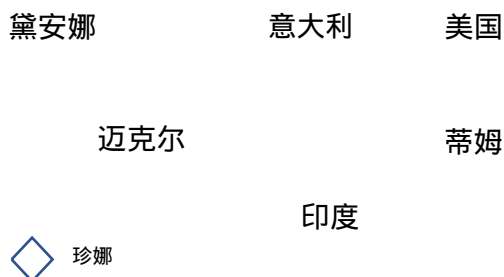


图3.1：个人的表示方式

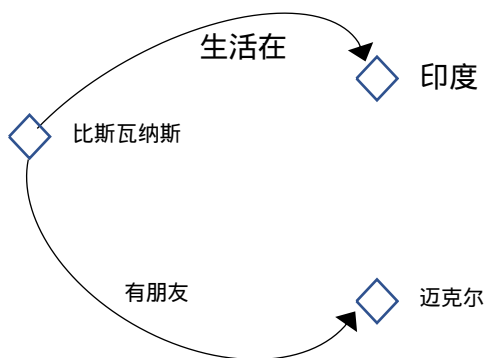


图3.2：属性的表示方式



个人也被称为实例。个人可以被称为类的实例。

3.1.2 属性

属性是个体之间的二元关系。也就是说，属性将两个人联系在一起。例如，财产朋友可能把比斯瓦纳斯个人和迈克尔个人联系起来，或者财产朋友可能把迈克尔个人和奥里安娜个人联系起来。属性可以是相反的。例如，孩子的逆是父的。属性可以被限制为具有一个值，即功能性。它们也可以是传递的或对称的。第4.8节详细说明了这些特性特性。图3.2显示了一些属性的表示形式。



属性类似于面向对象编程（OOP）中的属性。然而，在OWL和OOP的性质之间有重要的差异。最重要的区别是，OWL属性是独立于类而存在的第一类实体。我们鼓励OOP开发人员使用read：<https://www.w3.org/2001/sw/BestPractices/SE/ODSD/>

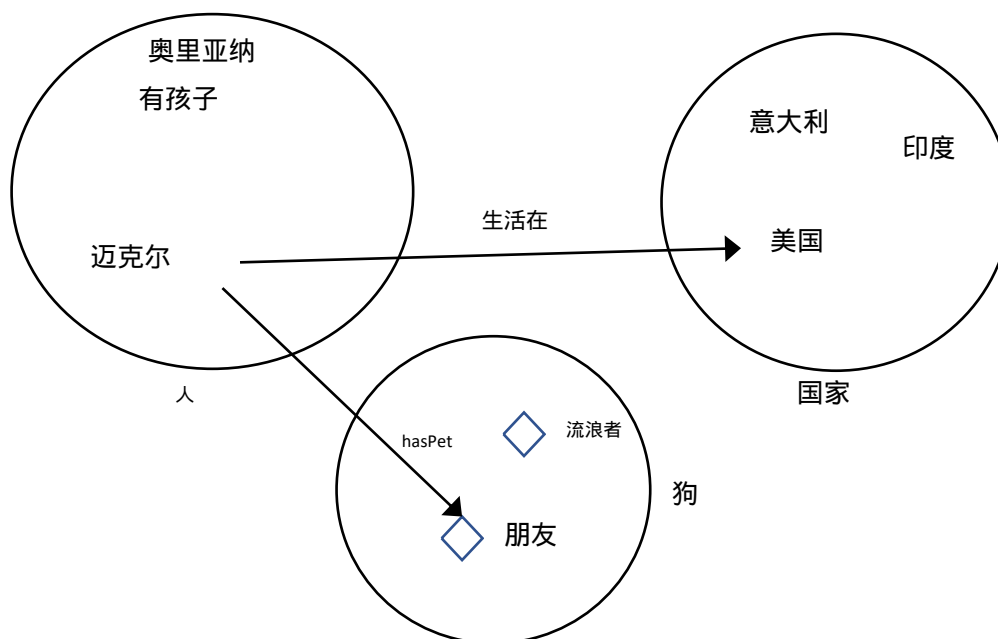


图3.3：包含个人的类的表示法

3.1.3 类

OWL类是包含个体的集合。它们使用正式的（数学）描述来描述，严格定义了类的成员需求。例如，猫类将包含我们感兴趣的领域中的所有属于猫的个体。2个类可以被组织成一个超类-子类的层次结构，这也被称为分类法。然而，分类法通常是树类的。即，每个节点只有一个父节点。OWL中的类层次结构并不局限于树，而且多重继承可以成为以直观的方式表示数据的强大工具。

子类专门化（也就是被包含）它们的超类。例如，考虑类动物和狗-狗可能是动物的一个子类（所以动物是狗的超类）。即所有的狗都是动物，所有的成员都是动物类的成员。OWL和Protege

² 个体可以属于多个类，而类可以有多个超类。与OOP不同，多重继承通常不可用或不受欢迎，这在OWL中很常见。

提供一种被称为描述逻辑或简称DL的语言。DL的一个关键特征是，这些超类-子类关系（又名包含关系）可以由推理机自动计算——稍后将详细介绍。图3.3显示了一些包含个体的类的表示——类用椭圆形表示，就像维恩图中的集合一样。

在OWL中，类可以由描述来构建，这些描述指定了个人必须满足的条件，这样它才能成为类的成员。如何制定这些描述将随着教程的进行而进行解释。

第四章建立OWL本体论

本章介绍了如何创建一个披萨的本体论。我们使用披萨是因为它几乎是每个人都熟悉的东西。

练习1：创建一个新的OWL本体论

1. 启动抗议活动。当Protege每天第一次打开时，它会显示一个包含所有可用插件的屏幕。你也可以在任何时候通过使用文件>检查插件。在本教程的此时点上，您将不需要任何插件，所以只需点击_ 现在没有按钮。
2. Protege的用户界面由几个标签组成，如活动本体、实体等。当您启动Protege时，您应该在“活动本体”选项卡中。这是关于整个本体的概述信息。Protege总是以一个新的无标题本体打开，您可以从它开始。你的本体应该有一个类似于：<http://www.semanticweb.org/yourname/ontologies/2020/4/untitled-ontology-的IRI> 27 编辑本体的名称（在本例中是最后一个“/”之后的部分-本体-27），并将其更改为类似于Pizzatu教程的内容。注意：下面所示的Pizza本体IRI（例如，图4.3）显示了我编辑了Protege为我生成的默认值后的IRI。您的IRI将看起来有所不同，并将基于您的名称或您的组织的名称。
3. 现在，您要保存新的本体。选择文件>Save。这应该会出现一个窗口：选择一种在保存“Pizzatu教程”本体时使用的格式。有一个下拉菜单的格式要使用。应该通过单击OK按钮来选择默认的RDF/XML语法。这将打开操作系统用于保存文件的标准对话框。导航到您想要使用的文件夹，然后输入文件名，类似于“披萨教程”，然后选择“保存”。



和任何你要处理的文件一样，定期保存你的工作是一个好主意，这样如果出了问题，你就不会失去你的工作。在教程中保存特别重要的某些地方，教程会提示您这样做，但经常保存工作是一个好主意，而不仅仅是在提示的时候。

下一步是设置一些与新实体的名称相关的首选项。请记住，比在Protege中，任何类、个体、对象属性、数据属性、注释属性或规则都被称为实体。OWL中的术语名称实际上可以指两个不同的概念。它可以是IRI³的最后一部分，也可以引用注释属性（通常是rdfs: label），用于为实体提供更用户友好的名称。我们将在下面第7章中详细讨论这个问题。现在，我们只想正确地设置参数，以便教程的未来部分（特别是关于SPARQL查询的部分）将适当地工作。

³ IRI 类似于URL。这将在下面的第7章中详细讨论。

练习2：设置新实体和渲染的首选项

1. 转到**临时程序**中的文件>首选项。这将带来一个新的窗口，里面有很多很多不同的标签。单击**“新建实体”**选项卡。这将打开一个看起来类似于图4.1的选项卡。该选项卡的顶部部分是一个标记为**实体IRI**的框。设置参数如图4.1所示。I.e.,从**活动本体IRI**开始。后面是**#**。以**用户提供的名称**结束。如果最后一个参数**设置为自动生成的名称**，则将其更改为用户提供的名称。这是最有可能不同的参数，但也要检查其他两个。
2. 现在，请选择**“渲染器”**选项卡。它应该如图4.2所示。最重要的是，检查**实体渲染是否设置为“按实体IRI短名称（ID）渲染”，而不是“按注释属性渲染”**。如果在这点上还不完全有意义，请不要担心。这里的问题有点复杂和微妙，所以我们将它们推迟到，直到您理解了OWL本体的基本概念之后。我们将在下面的第7章中讨论这些细节。现在，您只需要确保首选项设置得当，以配合本教程的其他部分。

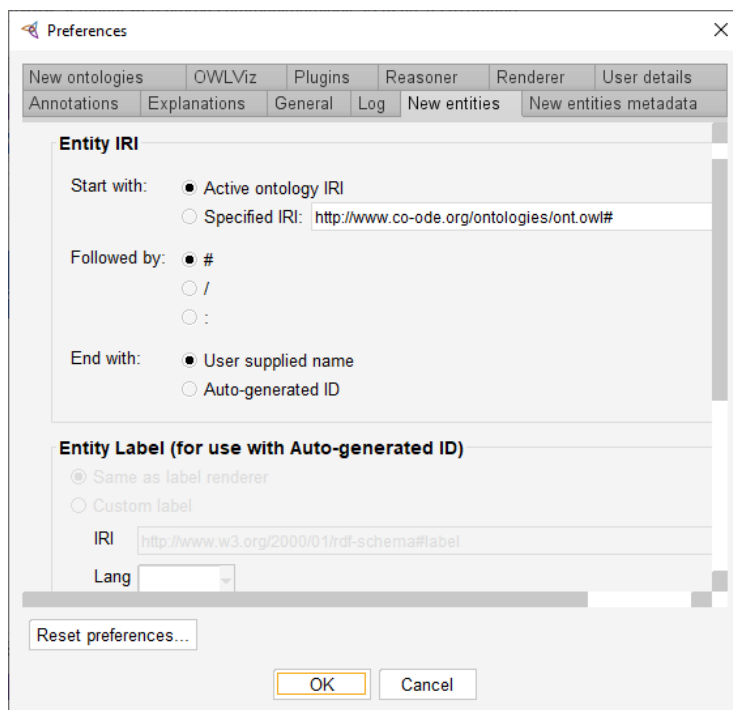


图4.1：“新建实体”选项卡

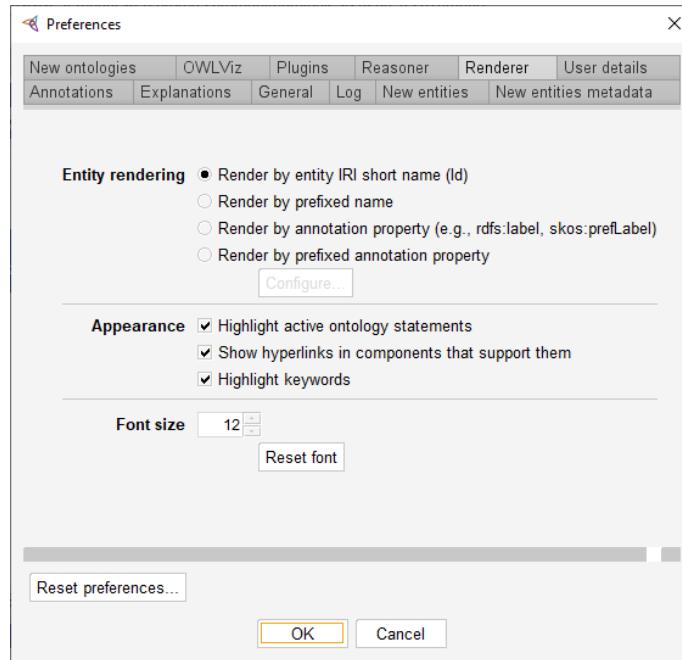


图4.2渲染器选项卡

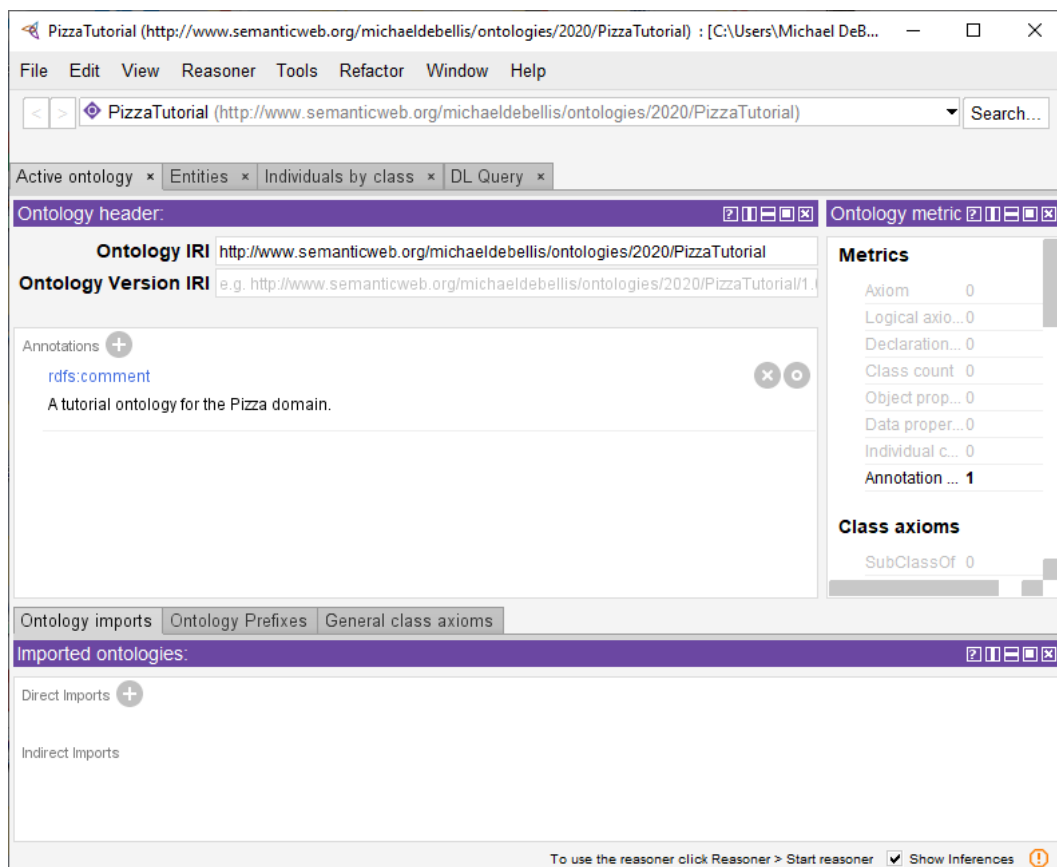


图4.3：带有新注释的“活动本体”选项卡

练习3：在 ontology 中添加注释

1. 请确保您在“活动的本体”选项卡中。在本体版本 IRI 和本体版本 IRI 字段下面的视图中，可以找到“注释”选项，并单击+符号。这将打开一个菜单，以在本体上创建一个新的注释。
2. 默认情况下，`rdfs:comment` 注释应该被突出显示。如果它没有被突出显示，请点击它。然后在右侧的视图输入一个新的注释。类似于披萨领域的教程本体。
3. 单击“确定”。您的活动本体选项卡应该像图4.3。



图4.4：类层次结构视图选项

4.1 命名的类

OWL本体的主要构建块是类。在Protege 5中，可以在实体选项卡中编辑类。“实体选项卡”有许多子选项卡。当您选择它时，默认值应该是类层次结构视图，如图4.5.4所示。所有空的 ontology 都包含一个名为`owl:Thing`的类。OWL类是一组个体类。类猫头鹰：`Thing`是代表包含所有个体的集合的类。正因为如此，所有的类都是猫头鹰的子类。

⁴ “实体”选项卡中的每个子选项卡也作为其自己的主要选项卡存在。在本教程中，我们将参考诸如“类层次结构”选项卡或“对象属性”选项卡等选项卡，决定用户是从“实体”选项卡访问它们，还是将它们创建为独立选项卡。

练习4：创建课程：披萨、披萨和披萨基地

1. 选择“类别”层次结构视图后，导航到“实体”选项卡⁵。确保猫头鹰：这个东西被选中了。
2. 按图4.4中所示的“添加子类”图标。此按钮将创建所选类的一个新的子类。在这种情况下，我们想创建一个猫头鹰的子类：Thing。
3. 这将出现一个名为“创建新类”的对话框，其中带有新类名称的字段。输入披萨，然后选择OK。
4. 重复前面的步骤添加类和确保猫头鹰：在使用添加子类图标之前选择了东西，这样你所有的类都是猫头鹰：东西的子类。您的用户界面现在应该如图4.5所示。不要担心有些类会用红色突出显示。这是因为推理机还没有运行。我们将很快讨论这个问题。

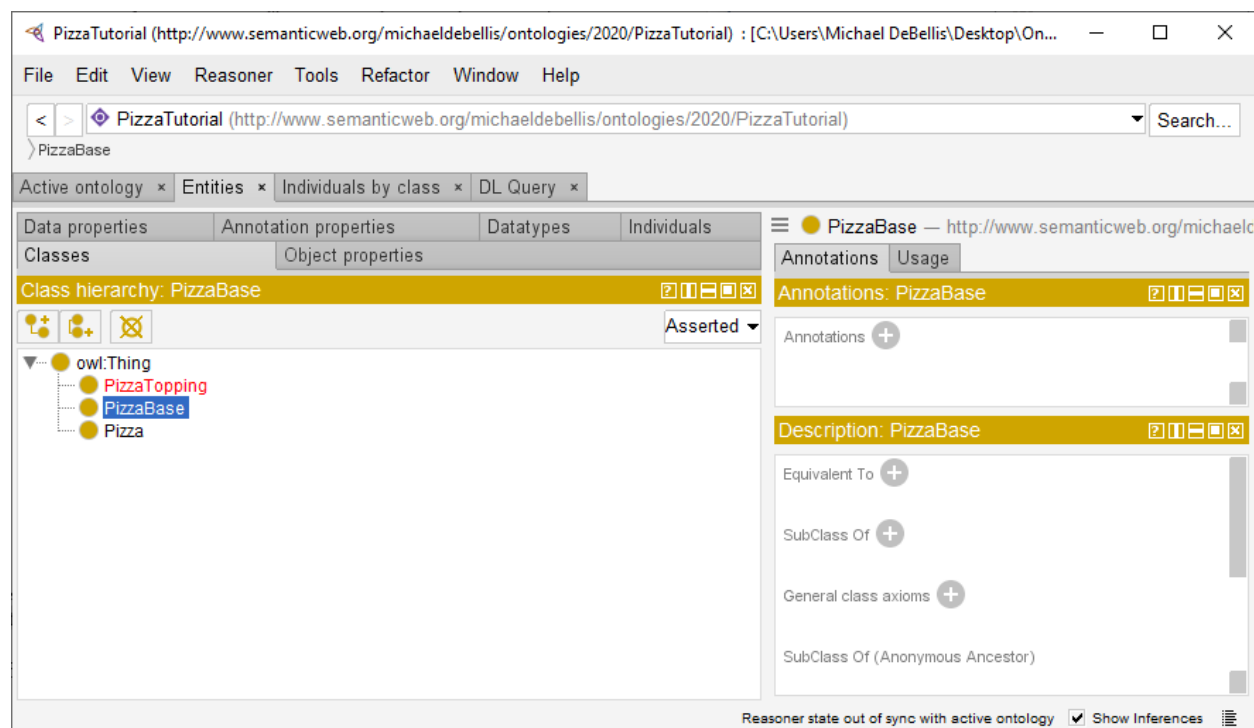


图4.5 “实体”选项卡中的类的子选项卡

⁵ “实体”选项卡是一个大选项卡，它包含诸如类、对象属性、数据属性等选项卡。作为子选项卡。每个子选项卡也是一个主要的选项卡（可从窗口>选项卡选项访问），可以自己创建。因为我在不同的时间拍摄屏幕快照，所以我并不总是完全一致的。有时我使用类作为实体选项卡的子选项卡，有时我单独作为主要选项卡。此外，在不同的时间点，我有其他标签打开，这取决于我正在做的其他工作。因此，您的UI看起来不会与数字完全相同。图中可能有其他选项卡不在你的UI中，反之亦然。



对于OWL实体，没有强制性的命名约定。在第7章中，我们将更详细地讨论名称和标签。最佳实践是选择一组命名约定，然后在整个组织中遵守该约定。在本教程中，我们将遵循以下标准，即类和个人名称以每个单词的大写字母开头，而不包含空格。这被称为骆驼背符号。例如：披萨饼，披萨饼，等等。此外，我们还将遵循类名总是单数而不是复数的标准。例如，披萨而不是披萨，披萨饼而不是披萨。

4.2 使用推理器

您可能会注意到，有一个或多个类用红色突出显示，如图4.5所示。这是因为我们还没有运行推理器，所以Protege无法验证我们的新类没有不一致。当只是在一个新的本体中创建类和子类时，不一致的可能性不大。然而，经常运行推理机是一个好主意。当存在不一致时，越早发现就越容易修复。新用户犯的一个常见错误是，进行大量开发，然后运行推理机，却发现存在多个不一致，这可能会使调试更加困难。所以，让我们养成经常运行推理机的好习惯。Protege附带了一些推理器，其他可作为插件。由于我们将在本教程中编写一些SWRL规则，所以我们希望使用颗粒推理器。在编写本教程时，它对SWRL有最好的支持。

练习5：安装和运行小颗粒推理器

1. 检查是否安装了颗粒推理器。单击推理器的菜单。在菜单的底部将会有有一个已安装的推理机的列表，如隐士和可能的颗粒。如果颗粒在该菜单中是可见的，然后选择它并跳过步骤3。
2. 如果小球是不可见的，然后做文件>检查插件，并从可用的插件列表中选择小球，然后选择安装。这将安装小丸，你应该得到一个消息，它将生效时，你启动宝门。做一个文件>保存保存你的工作，然后退出项目并重新启动它。然后进入文件>打开最近。您应该可以在最近的列表看到您保存的披萨教程。选择它以加载它。现在你应该看到颗粒在推理器菜单下，并能够选择它，这样做。
3. 在推理器菜单中选择颗粒后，执行命令推理器>启动推理器。推理器应该运行得非常快，因为本体论非常简单。您会注意到，程序窗口右下角的小文本消息已经更改为“现在说推理器活动”。下次当您对本体进行更改时，文本将被更改为：推理器状态与活动本体不同步。对于小的本体，推理机运行得非常快，并且养成经常运行它的习惯是一个好主意，就像在每次更改之后一样。
4. 在运行推理器后，一个或多个类可能仍然会用红色突出显示。如果发生这种情况，请执行：Window>刷新用户界面，任何红色高光都应该消失。当您的用户界面似乎显示了一些东西时，您并不期望要做的第一件事就是尝试这个命令。

5. 我们想做的最后一件事是配置推理器。默认情况下，推理器不会执行所有可能的推断，因为对于大型和复杂的本体，一些推断可能需要很长时间。在本教程中，我们将始终处理小而简单的本体，所以我们希望看到推理机所能做的一切。转到：Resoner>配置。这将出现一个对话框，其中包含推理器可以执行的几个推理复选框。如果没有全部检查一下，那就全部检查一下。您可能会收到一个警告，说一些推论可能需要很多时间，但您可以忽略这些时间，因为您的本体将会很小。

4.3 不相交的类

在将类添加到Pi zzabase本体之后，我们现在想说这些类是不相交的。也就是说，没有一个人可以是这些类的一个实例。在集合理论的术语中，这三个类的交集是空集合：猫头鹰：没有。

练习6：让披萨饼、披萨饼和披萨饼相互分离

1. 在类层次结构中选择类Pi zza。
 2. 在“描述”视图找到“不相交与”选项，并选择它旁边的(+)符号。见图4.6中的红色圆圈。
 3. 这应该会打开一个带有两个选项卡的对话框：类层次结构和表达式编辑器。您现在需要类层次结构（我们稍后将使用表达式编辑器）。这为您提供了一个接口来选择与类层次结构视图相同的类。使用它可以导航到Pi zzaBase。按住换档键，然后选择夹层基地和夹层顶部。选择“确定”。
 4. 做一个推理器，>同步推理器。然后看看披萨基地和披萨顶部。您应该看到，它们每个类都定义了适当的不相交公理，以表明这些类与其他两个类是不相交的。
-

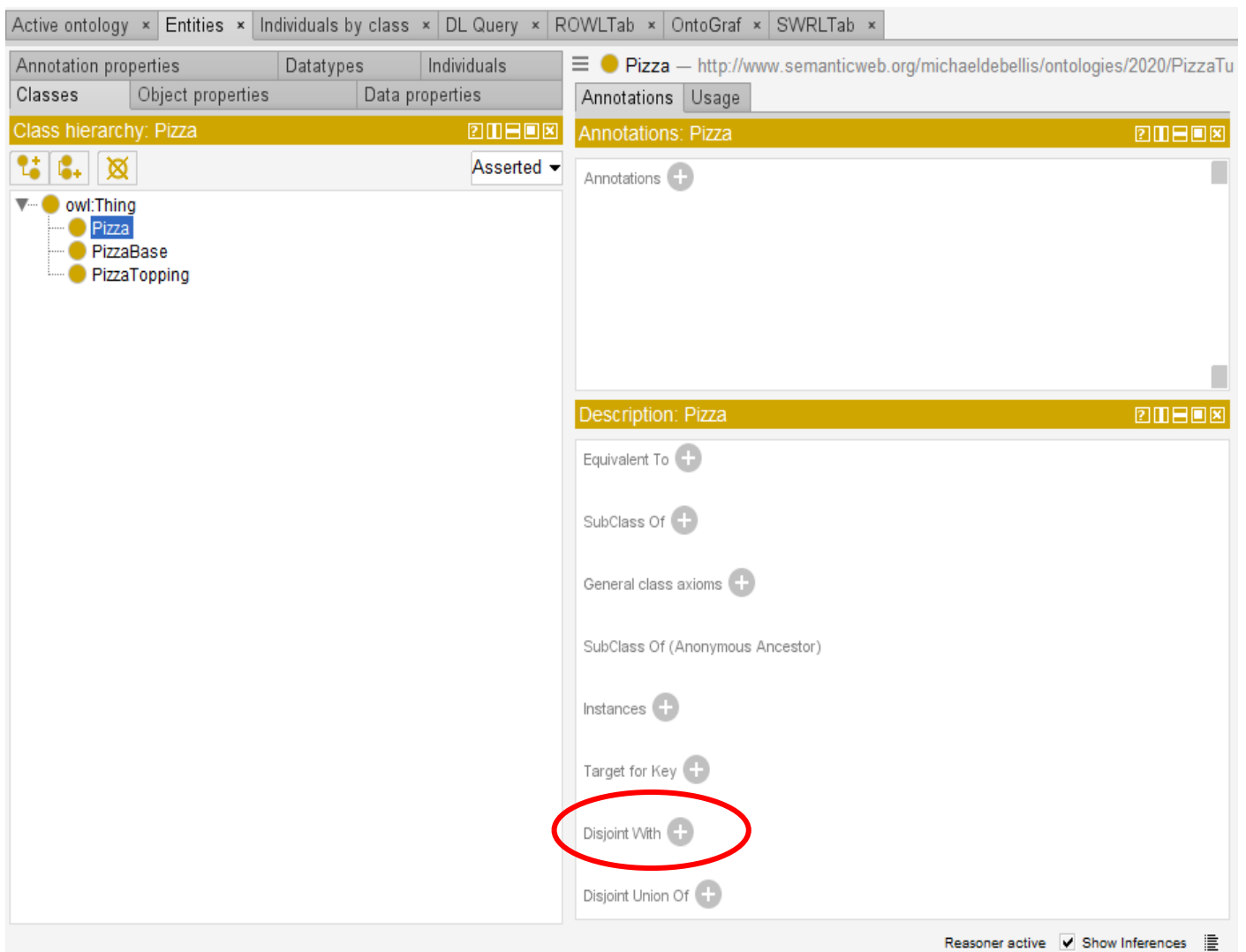


图4.6：类描述视图中的不相交选项



假设OWL类是重叠的，即，默认情况下它们不是不相交的。这通常很有用，因为在OWL中，与大多数面向对象的模型不同，多重继承并不受鼓励，而且可以成为建模数据的强大工具。如果我们希望类是不相交的，那么我们必须明确地声明它们是不相交的。这通常是一个很好的开发策略，从不是不相交的类开始，然后一旦模型更充实，就使它们不相交，因为哪些类从一开始并不总是明显是不相交的。

4.4 使用创建类层次结构

在本节中，我们将使用工具>创建类层次结构来同时创建多个类。

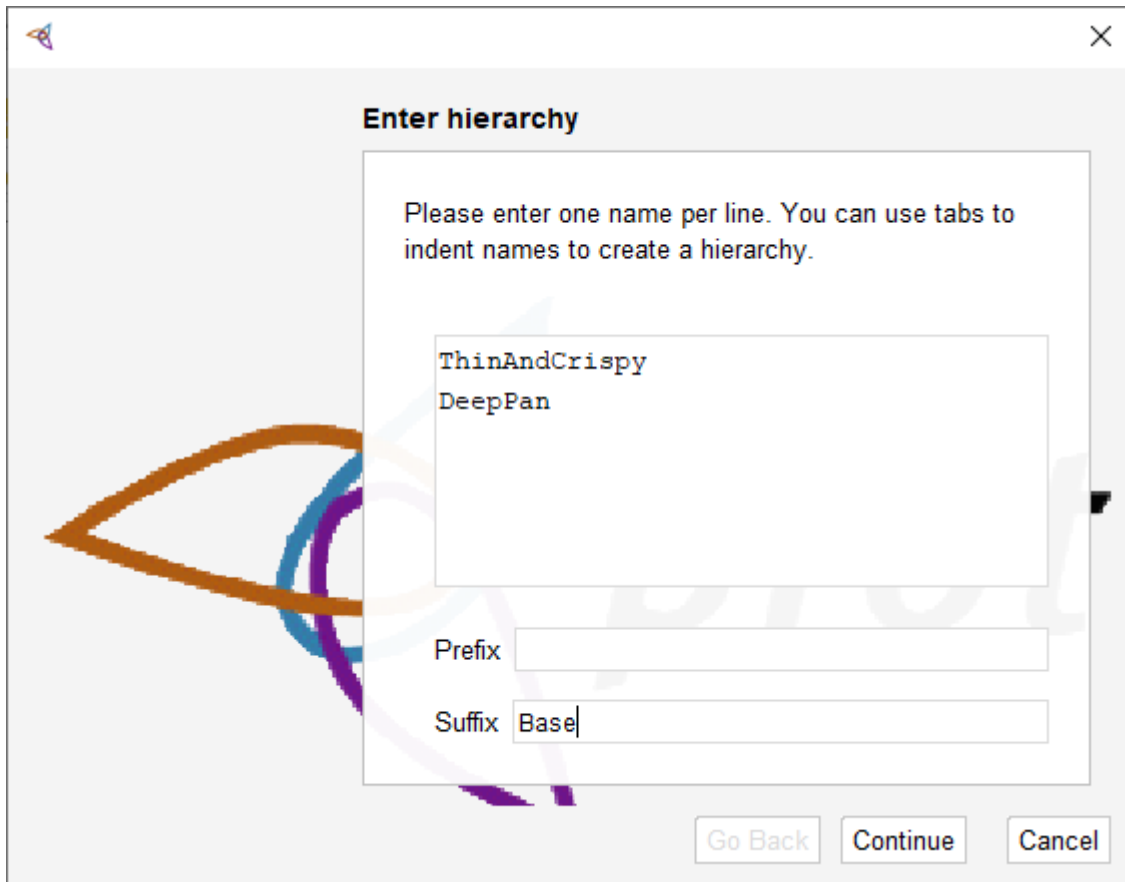


图4.7：创建类层次结构向导

练习7：使用“创建类层次结构”工具来创建Pi zzaBase的子类

1. 在类层次结构中选择类Pi zzaBase。
2. 选择了该数据库后，请使用工具>创建类层次结构菜单选项。
3. 这将打开一个向导，使您能够一次创建一个嵌套的类组。您应该看到一个标记为“Enter层次结构”的窗口，您可以在每行上输入一个名称。您还可以使用选项卡键来指示一个类是它上面的类的子类。现在，我们只想输入披萨基地的两个子类：深度交叉基地和深度面板基地。向导所做的一件事是自动为我们添加前缀或后缀。所以只要进入薄脆，点击返回，进入深盘。然后在后缀字段中添加基数。您的窗口应该如图4.7所示。
4. 选择“继续”。这将把您带到一个窗口，询问您是否想使兄弟姐妹类不相交。应该检查默认值（使它们不相交），这是我们在这种情况下想要的（底座不能既是深的，也不是薄的），所以只要选择“完成”。同步推理器。您的类层次结构现在应该如图4.8所示。

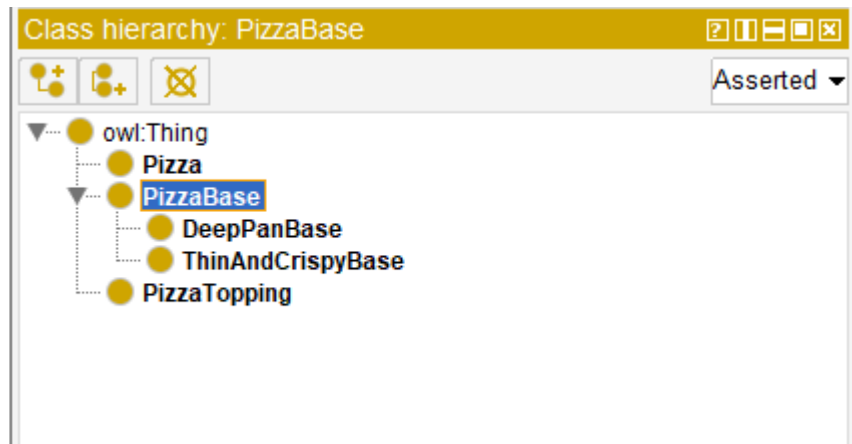


图4. 8：新的类层次结构

4.5 创建Pizzatock层次结构

我们将再次使用Tools>创建类层次结构，但这次是为了创建一个更有趣的层次结构，并使用额外的子类来建模披扎顶点的子类。

练习8：创建Pizza顶部的子类

1. 在类层次结构中 选择类顶部。
2. 选择披扎时，使用工具>创建类层次结构菜单选项。
3. 这将再次打开向导。我们希望我们所有的顶部都以顶部结束，所以在后缀字段中输入顶部。然后创建嵌套结构，如图4.9所示的结构。使用Tab键在需要时缩进类。
4. 选择“继续”。这将把您带到询问是否您想使兄弟姐妹类不相交的窗口。我们确实想要这样，所以选中并单击完成。同步推理器。现在，您的类层次结构应该如图4.10所示。

Enter hierarchy

Please enter one name per line. You can use tabs to indent names to create a hierarchy.

```
Cheese
  Mozzarella
  Parmesan
Meat
  Ham
  Pepperoni
  Salami
  SpicyBeef
Seafood
  Anchovy
  Prawn
  Tuna
Vegetable
  Caper
  Mushroom
  Olive
  Pepper
    RedPepper
    GreenPepper
    JalapenoPepper
  Tomato
```

Prefix

Suffix

图4。9使用创建类的层次结构来创建顶峰子类

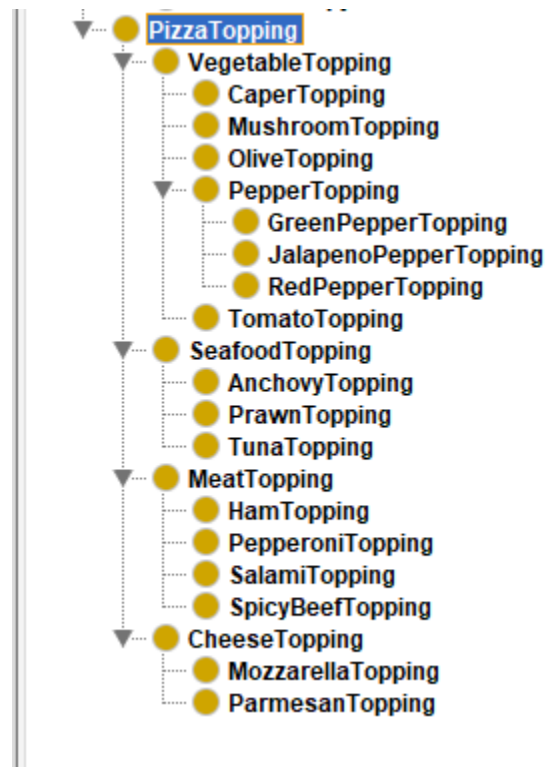


图4。10. 新的顶级班级等级制度



到目前为止，我们已经创建了一些简单的命名类和子类，希望它们看起来很直观和明显。然而，作为OWL中的一个子类到底意味着什么？例如，素食辣酱成为披扎辣酱的一个子类意味着什么？在OWL子类中意味着必要的含义。也就是说，如果素食顶点是披扎区顶点的一个子类，那么所有的素食顶点的实例也是披扎区顶点的实例。正是出于这个原因，我们尝试制定标准，比如所有的顶峰课程都以“顶峰”结束。否则，我们似乎是在说，任何一种火腿，就像三明治里的火腿，都是一种馅饼酱或披扎酱，这不是我们的意思。对于大型本体，严格注意类和其他实体的命名可以防止潜在的混淆和错误。

4.6 OWL属性

OWL属性表示关系。有三种类型的属性，对象属性、数据属性和注释属性。对象属性是两个个体之间的关系。数据属性是单个数据类型和数据类型之间的关系，如xsd:字符串或xsd:dateTime。注释属性通常也具有数据类型作为值，尽管它们可以有对象。注释属性通常是元数据，如注释或标签。在OWL中，只有个人可以拥有对象和数据属性的值，但任何实体都可以拥有注释属性值，因为元数据应用于所有实体。注释属性通常无法进行推理。例如，我们稍后将讨论的SWRL规则不能查看或更改注释属性的值。在本章中，我们将重点介绍对象属性。数据属性详见第5章。在本教程的当前版本中，我们只讨论了注释属性rdfs:label（参见第7章），但是它们相当直观。

可以使用图4.11中所示的“实体”选项卡中的“对象属性”子选项卡来创建属性。就像所有的OWL类最终都是猫头鹰的子类一样，所有的属性最终都是OWL的子属性。子属性与子类类似，只是它是关于属性中的元组的。例如，父将是父的子属性，因为父中的所有元组都在父中，但反之亦然。例如，如果萨沙是巴拉克的父亲，那么她也是巴拉克的父母。然而，她也是米歇尔的父母，但她的父亲却不是米歇尔。相反，她的母亲是米歇尔，也就是说，这个母亲也是她的父母的一个子财产。

用于输入属性的GUI也类似于用于输入类的GUI。包含一个框下的另一个框的第一个图标将创建所选属性的子属性。第二个图标显示同一级别的两个框，将为所选属性创建一个兄弟属性，而带有X的图标将删除所选属性。

练习9：创建一些属性

1. 选择“实体”选项卡中的“对象属性”子选项卡（请参见图4.11）。
2. 确保选择了顶级对象属性。单击左侧的嵌套框图标，以创建owl的新子属性：top对象属性。当提示输入当前成分中的新属性类型的名称时。
3. 您可以使用向导创建多个类，也可以使用一个来创建多个属性。选择其中一个成分，然后选择工具>创建对象属性层次结构。输入新的属性名称，包括顶部和base。选择“继续”并接受对象属性不相交的默认值。
4. 同步推理器。您的窗口现在应该如图4.11所示。



对于那些熟悉实体-关系模型的人，OWL对象属性类似于关系，而数据属性类似于属性。对象属性类似于OOP中具有某个类范围的属性，而数据属性类似于具有数据类型范围的OOP属性。

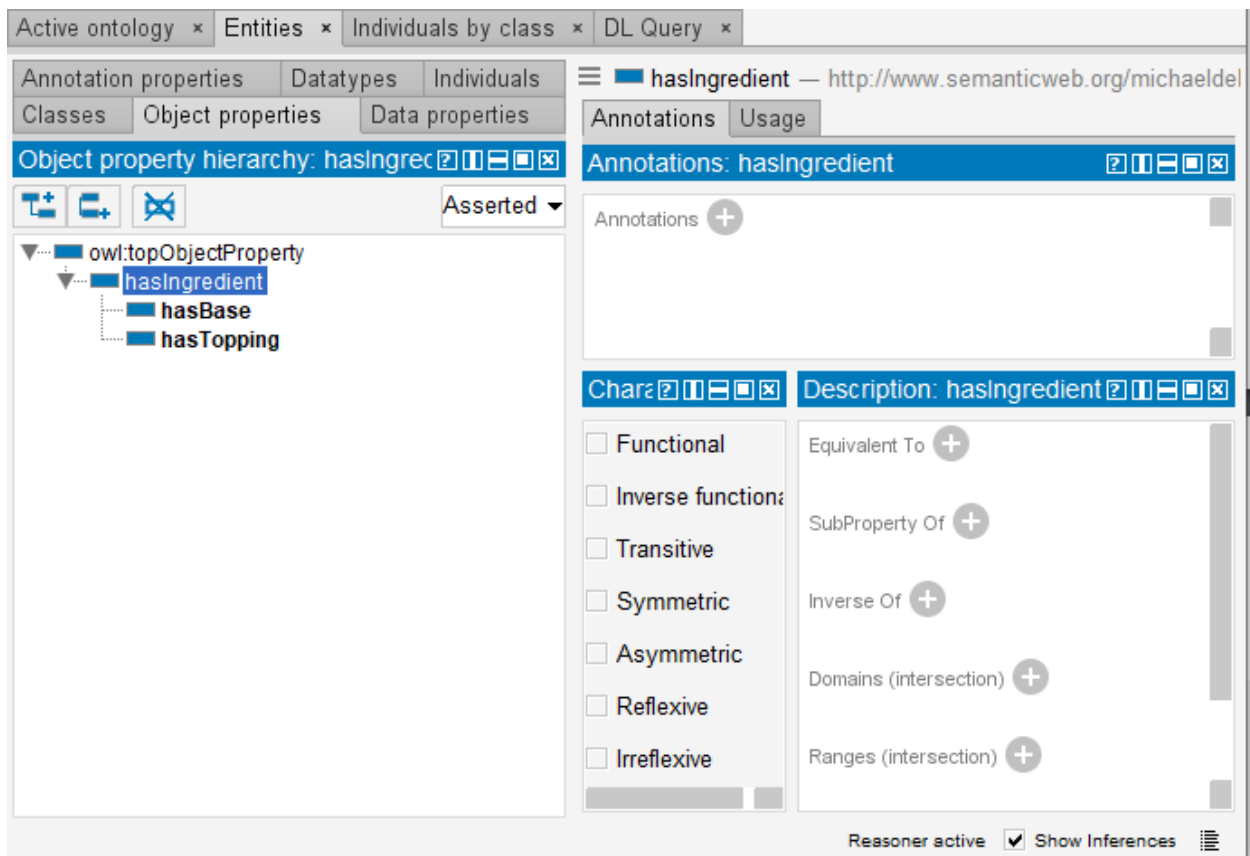


图4.11添加了一些对象属性

4.7 逆属性

每个对象属性都可以具有相应的逆属性。如果某些属性将个体a链接到个体b，那么它的逆属性将个体b链接到个体a。例如，在图3.3中，个人迈克尔的宠物伙伴。在这个例子中，hasPet是一个对象属性，它从一个人映射到他们的宠物，这被称为该属性的域和范围。迈克尔是人类的一个例子，而巴迪是宠物类的一个例子。这个宠物的财产从一个人指向那个人的宠物。相反的属性可以是属性，它将由两个走另一条路的人之间的联系来表示，从巴迪到迈克尔。只要有可能，最好使用属性来遵守这种类型的命名标准。在一个方向上移动的属性是属性，它们的逆方向是属性。

练习10：创建一些逆属性

1. 使用“对象属性”选项卡可以创建一个名为“成分o”的新对象属性（这将是“成分”的逆属性）。确保成分是一个兄弟属性，如果有成分和一个子属性。
2. 单击“说明”视图中“逆”旁边的“添加”图标（“+”）。您将看到一个窗口，其中显示所有当前属性的嵌套视图。选择某一个成分，使其与成分的倒数相反。

3. 选择是的成分，然后选择工具>创建对象属性层次结构。输入顶部，然后在新行上输入基础of。与前面一样，选择“继续”，并未选中不相交属性，然后选择“完成”。重复步骤2，使顶部的倒数和底部的倒数倒数。

4. 同步推理器。您的窗口现在应该如图4.12所示。

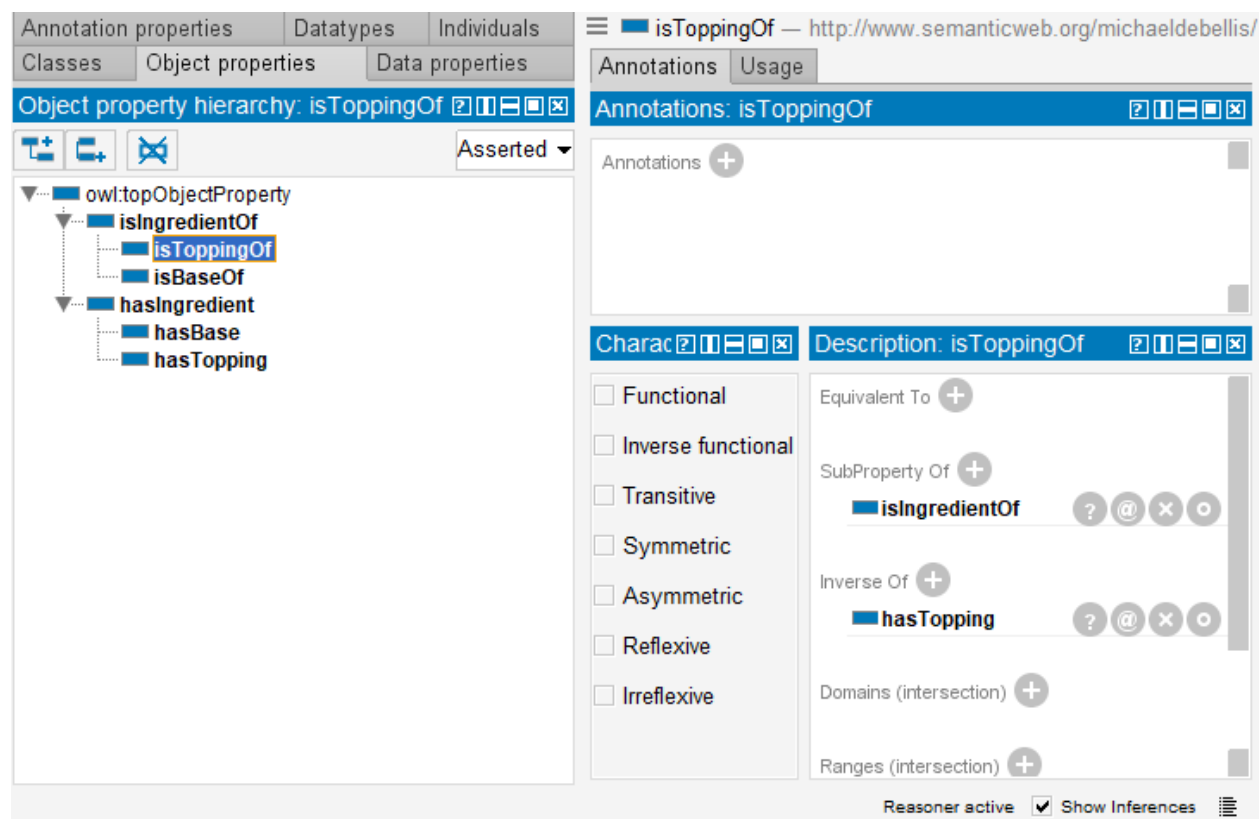


图4.12逆属性

4.8 OWL对象属性特征

OWL允许通过使用属性特性来丰富属性的意义。下面的几节将讨论属性可能具有的各种特性。如果你熟悉集合理论中关系的基本概念，那么这些特征就已经为你所熟悉了。在图4.12中，您可以看到特性：查看属性的复选框列表：函数、逆函数、传递等。

4.8.1 功能属性

如果一个属性是功能性的，那么对于一个给定的个人，最多可以有一个个体通过该属性与该个人相关联。例如，财产有生母——一个人只能有一个生母。如果我们说琼的生母是佩吉，我们也说琼的生母是玛格丽特，那么因为生母是一个功能属性，我们可以推断佩吉和玛格丽特一定是同一个人。这可能会发生在OWL中，因为

与许多语言不同，它没有唯一的名称假设。除非另有特别说明，推理者可以推断出两个名字不同的人实际上是同一个人。然而，应该指出的是，如果佩吉和玛格丽特被明确地声明为两个不同的个体，那么上述陈述将导致推理者推断出在本体论中存在不一致性。我们将在第7章中更多地讨论名字。

在第4.16节中，我们将讨论在性质上的基数限制。例如，自行车类的带轮属性至少有2个（允许训练轮），而独轮车类的带轮被定义为1。一个函数属性等价于一个具有基数限制的属性，即它的最大值为1。泛函来自数学，其中函数定义为一个关系，其中域的每个成员最多有一个值。例如，多个关系不是函数的，因为对于任何数 X ，多个（实际上是无限数）可以是多个 X ，但多个关系是函数的，因为对于任何数 X ，多个关系总是产生一个唯一的值。

4.8.2 反函数性质

如果一个属性是逆函数的，那么它就意味着这个逆属性是函数的。对于一个给定的个人，最多可以有一个个人通过该财产与该个人相关联。根据第4.8.1节的例子，生母的逆值应该是生母。财产的出生母亲将不会正常运作，因为一个女人可以是几个孩子的生母。然而，它将是相反的功能，因为每个人都有一个人母亲。

4.8.3 瞬态特性

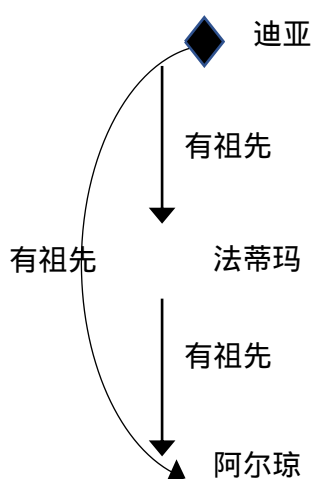


图4.13瞬态属性

如果一个属性 P 是传递的， P 将个体 a 与个体 b 相关，个体 b 也与个体 c 相关，那么我们可以通过属性 P 推断个体 a 与个体 c 相关。例如，图4.13显示了传递属性与祖先的一个例子。如果个体迪亚人的祖先是法蒂玛人，而法蒂玛人的祖先是阿尔琼人，那么我们就可以推断迪亚人的祖先是阿尔琼人——这如图4.13中的曲线所示。

数学中传递性质的一个例子是 $>$ 关系。如果 $x > y$ 和 $y > z$ ，那么 $x > z$ 。

请注意，如果一个属性是可传递的，那么它就不能是函数的。另外，如果一个性质是传递的，那么它的逆性质也必须是传递的。例如， $>$ 的逆是 $<$ ， $<$ 也是传递的。我们将在第6章中看到一个例子。

4.8.4 对称的和不对称的性质

如果一个属性 P 是对称的，并且该属性将个体 a 与个体 b 相关，那么个体 b 也通过属性 P 与个体 a 相关。兄弟属性或配偶就是的例子

对称性质。如果米歇尔支持巴拉克，那么巴拉克就是米歇尔。一个对称的性质是它自身的逆性质。

不对称属性是一个永远不能具有对称值的属性。If a属性P是不对称的，那么如果a与b相关，那么属性b不能与a相关。不对称属性的一个例子是它有生母。如果迪娅的生母是法蒂玛，那么法蒂玛就不可能是迪娅的生母了。

4.8.5 反射特性和不反射特性

自反属性是一种总是将个体与个体自身联系起来的属性。如果一个性质P是自反的，那么对于所有的个体，一个P总是把a和a联系起来。相等是一个自反性质的最常见的例子。对于任何对象a，a总是等于a。一个不可反射的属性是.....你猜对了.....一个永远不能把个人与自己联系起来的属性。生母是一个不反射的例子，因为没有人不能成为自己的母亲。注意：您应该小心地使用反射性属性。自反性质的域总是猫头鹰的。原因很复杂，更多细节请参阅参考书目中的W3C Owl 2规范。重要的是，如果你做了一个属性自反，这意味着它的域是猫头鹰：东西。例如，如果你有一个自反属性，并声明它的域是某个类，比如人，推理者将推断出这个人等同于猫头鹰：可能会导致问题的东西。

4.8.6 推理器会自动强制执行属性特征

使用Protege的推理器会自动强制执行上面描述的所有特征。例如，如果用户输入迪亚的出生母法蒂玛和出生母的倒数，推理者将推断法蒂玛是迪亚的出生母。这些类型的特征可以显著减少用个人数据填充本体所需的工作量。

4.9 OWL属性域和范围

属性可以定义了一个域和范围。这些术语在OWL中和在数学和集合理论中具有相同的意义。属性的域是可以就此其断言该属性的所有对象的集合。范围是可以作为属性值的所有对象的集合。域和范围都是可选的。一般来说，定义它们是一个好主意，因为这样做可以在定义模型时捕获建模错误，而不是在试图使用它时在运行时。对象属性的域必须始终是一个类。对于数据属性，范围是一个简单的数据类型，如xsd：十进制。Protege中已经存在最常见的预定义数据类型。也可以定义新的数据类型，尽管大多数用户很少需要这样做。对于大多数情况，如果您正在考虑定义一个新的数据类型，则可能应该考虑将属性转换为对象属性，并将类定义为范围。对于熟悉实体-关系建模的人来说，对象属性类似于关系，而数据属性类似于属性。对于那些熟悉集合理论的人来说，一个性质与集合理论中的二元关系是相同的。

例如，在我们的披萨本体中，属性的顶点会将属于披萨类的个人与属于pizza顶点类的个人联系起来。顶部的领域是披萨，范围是披萨。逆属性有它们的域和范围的交换。在这个例子中，顶部的逆将被称为“顶部”。因此，isTopping0的域是顶部的范围（披萨顶部），而is顶部的范围是顶部的域（披萨）。

练习11：定义has顶点属性的域和范围

1. 导航到“对象属性”选项卡。选择hasTopping属性。
2. 单击“描述”视图中“域（交集）”旁边的“添加”图标（+）。您将看到一个显示几个选项卡的窗口。有多种方法来定义域和范围。现在，我们将使用最简单的方法（也是最常用的方法）。选择“类别层次结构”选项卡。然后从类层次结构中选择Pizza。你的UI应该如图4.14所示。单击确定。您现在应该在“描述”视图中看到域下方的披萨。
3. 重复步骤2，但这次从使用“描述”中的范围（交集）旁边的（+）图标开始。这次选择类的顶峰作为范围。
4. 同步推理器。现在选择是顶部的。您应该看到顶部的域和范围已由推理器填写（参见图4.15）。由于这两个属性是逆的，推理者知道一个的域是另一个的范围，反之亦然。这是为什么频繁运行推理机可以节省时间并帮助维护有效模型的另一个例子。请注意，这些值将用黄色突出显示。由推理者而不是由用户提供的任何信息都会以这种方式突出显示。

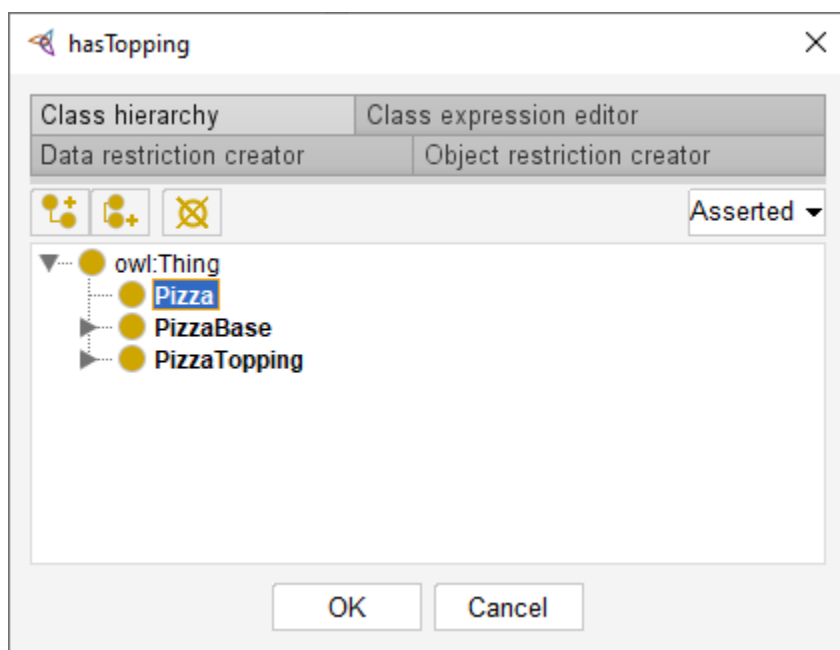


图4.14定义顶部的域

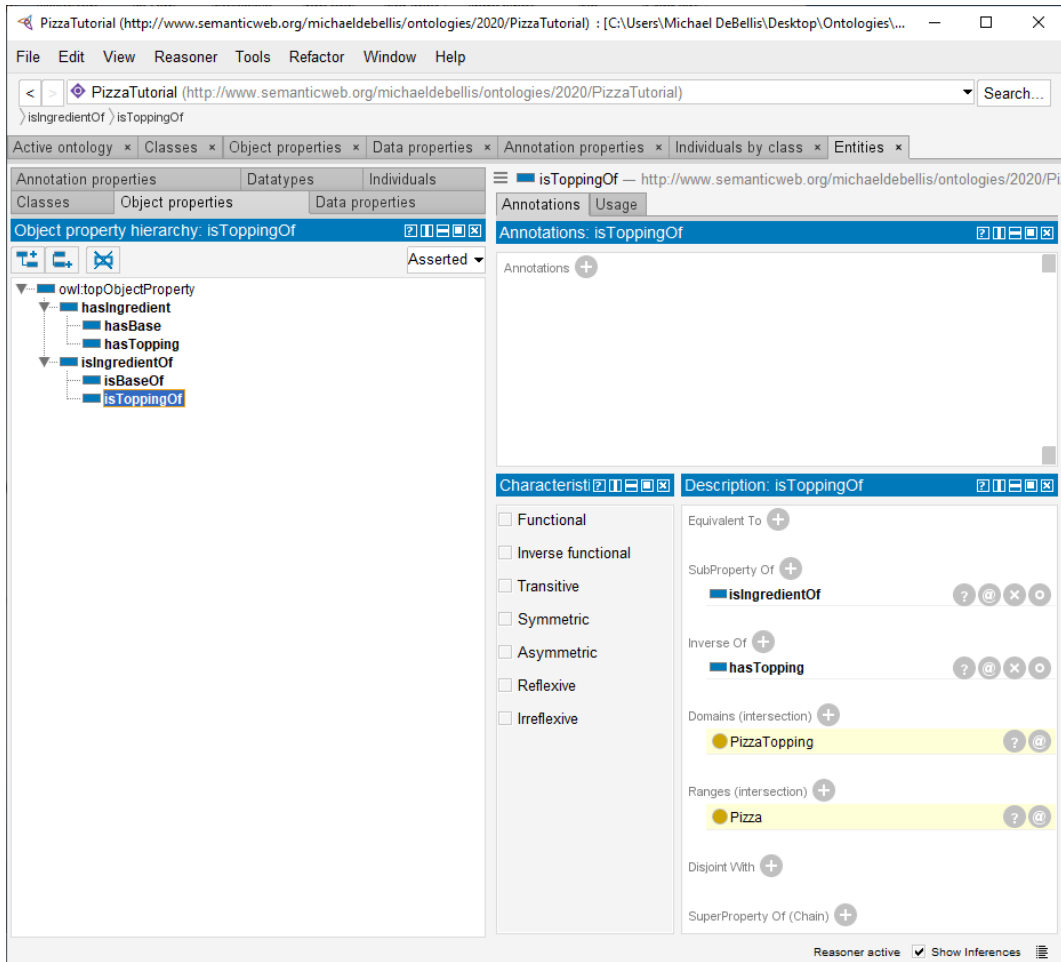


图4. 15由推理器推断出的域和范围



可以将多个类指定为属性的域或范围。新用户最常见的错误之一是这样做，并期望生成的域/范围是这两个类的并集。但是，请注意，在**描述视图**中的**域和范围**旁边说（交集）。这是因为有2个或更多的类作为域或范围的语义是这些类的交集，而不是并集。例如，如果一个定义的域属性披萨，然后添加另一个域奶油这意味着领域的属性必须是一个实例的披萨和冰淇淋不（人们经常期望）这两组的联盟将类披萨或类冰淇淋。另外，请注意，域和范围是用于推断的，它们不是数据完整性约束。这种区别将在SHACL的一节中详细解释。

练习12：为hasBase属性定义域和范围

1. 现在，我们将重复与上一个练习中相同的活动，但对于另一个属性：hasBase。请确保您仍然在“对象属性”选项卡上。选择hasBase属性。
 2. 单击hasBase的“描述”视图中“域（交集）”旁边的“添加”图标（+）。选择“类别层次结构”选项卡。然后从类层次结构中选择Pizza。
 3. 重复步骤2，但这次从使用has数据库说明中的范围（交集）旁边的（+）图标开始。这次选择类PizzaBase作为范围。
 4. 同步推理器。现在选择isBaseOf，您应该看到它的域和范围已经由推理器填写了。
-

4.10 描述和定义类

现在我们已经定义了一些属性，我们可以使用这些属性来定义一些更有趣的类。在OWL中有3种类型的类：

1. 原始类。这些类是由条件定义的，这些条件对于作为该类或其子类的任何个体的实例都是必要的（但不是充分的）。条件可以很简单：A类是B类的一个子类。首先，我们将首先定义基本类，然后再定义类。当推理机遇到一个原始类的实例时，它推断为该类定义的所有条件必须适用于该个体。
2. 定义的类。这些类是由必要条件和充分条件定义的。当推理机遇到一个满足一个已定义类的所有条件的个体时，它将推断出该个体是该类的一个实例。推理器还可以使用在类上定义的条件来更改类层次结构，例如，推断类A是B类的一个子类。我们将在后面的教程中看到这个例子。
3. 匿名类。这些课程你不会遇到太多，也不会遇到太多。在本教程中讨论了很多，但是很高兴了解它们。它们是由推理器在您使用类表达式时创建的。例如，如果您将属性的范围定义为Pizza顶点或PizzaBase，那么推理器将创建一个匿名类，表示这两个类的交集。

4.10.1 财产限制

在OWL中，属性定义了与一阶逻辑中的二进制关系具有相同语义和特征的二进制关系。有两种类型的OWL属性用于描述一个域：对象属性和数据属性。对象属性具有类作为其域和范围。数据属性有类作为它们的域，而有简单的数据类型，如xsd：字符串或xsd：dateTime作为它们的范围。在图3.3中，迈克尔个人与美国个人的财产关系。考虑所有的人谁是一个实例，也有相同的关系，每个人都生活在美国。这个组是一个集合或OWL类，如USARisirs。在OWL中，可以通过描述类中所有个体所持有的各种属性和值来定义一个类。这种定义在OWL中被称为限制。

以下是我们可能希望通过属性限制定义的个体类的例子：

至少有一种儿童关系的个体类别。

有2个或2个以上有孩子关系的个人类别。

与莫扎里拉奶酪的人至少有一种亲密关系的个体，也就是说，至少有马苏里拉奶酪的人构成顶部的东西

这一类是吃披萨的人，他们只与素食披萨类（即素食披萨）有惊人的关系。

在OWL中，我们可以使用限制条件来描述上述所有的类。OWL限制主要分为三类：

1. 量化器限制。这些属性描述了属性必须具有特定类的某些或所有值。
2. 基数限制。这些描述了必须通过特定属性与类相关的个体的数量。
3. has值限制。这些值描述了属性必须具有的特值。

我们将首先使用量词限制。量词限制可以进一步归类为存在性限制和普遍限制⁶。这两种类型的限制将在本教程中通过示例进行说明。

存在性限制描述了沿着指定属性参与至少一个关系的个体类别。例如，至少有一个（或一些）的个体与素食测试实例有显著关系。在OWL中，有些关键字被用来表示存在性的限制。

通用限制描述了对于给定属性，只有属性与特定类成员的个体的关系。例如，只有与素食类的关系。在OWL中，它们的关键字仅用于通用限制。

让我们仔细看看存在限制的例子。这是一个存在限制（如一些关键字所示），它限制了has的属性，并且有一个填充限制。这个限制描述的是与属于成员的个体至少有一种顶部关系的个体。



一个限制总是用来描述一个类。有时（我们很快就会看到）它可以是一个已定义的类。其他时候，它可能是一个匿名的类。在所有情况下，类都包含满足限制的所有个体，即，所有拥有成为类成员所需的关系的个体。在第9.2节中，其中一个SPARQL查询将返回几个匿名类。

⁶ 这些与一阶逻辑中的存在量化和普遍量化具有相同的意义。

系统将使用图4.17中所示的类描述视图来显示和编辑类的限制。类描述视图包含了用于描述类的大部分信息。类描述视图是一种描述和定义类的强大方法。这是在OWL中描述类和在其它模型中，如大多数面向对象的编程语言之间最重要的区别之一。在其他模型中，没有正式的定义来描述为什么一个类是另一个类的子类，而在OWL中则有。实际上，OWL分类器实际上可以根据用户定义的逻辑限制来重新定义类层次结构。我们将在后面的教程中看到一个示例。



限制在OWL中也被称为公理。这与在逻辑上有相同的意义。公理是由用户定义的，而不是由推理者推导出来的逻辑公式。如上所述，在Protege中，所有公理都以普通字体显示，而推理者推断的所有推断都用黄色突出显示。

4.10.2 存在性限制

存在限制描述了沿着指定属性与作为指定类或数据类型的成员的个体至少有一个（一些）关系的一个个体类。例如，一些披萨库描述了所有至少有一个关系的个体——用更自然的英语，所有至少有一个披萨库的个体。

练习13：向披萨添加一个限制，以指定披萨必须具有PizzaBase

1. 从“类”选项卡上的类层次结构中选择“披萨”。
2. 单击“披萨的说明”视图中“子类别”字段旁边的“添加”图标（+）。
3. 这将打开一个新窗口，其中有几个选项卡选项来定义一个新的限制。选择对象限制创建者。此选项卡左侧为“受限制”属性，右侧为“限制填充”属性。
4. 展开左侧的属性层次结构，并选择hasBase作为要限制的属性。然后在右边的限制填充物中选择类PizzaBase。最后，底部的限制类型应该设置为一些（存在的）。这应该是默认的，所以你不应该改变任何东西，但再次检查这是这种情况。你的窗口现在应该像图4.16。
5. 当你的UI看起来像图4.16时，单击OK按钮。这样应该会关闭窗口。运行推理机以确保事情一致。现在，您的主窗口应该如图4.17所示。

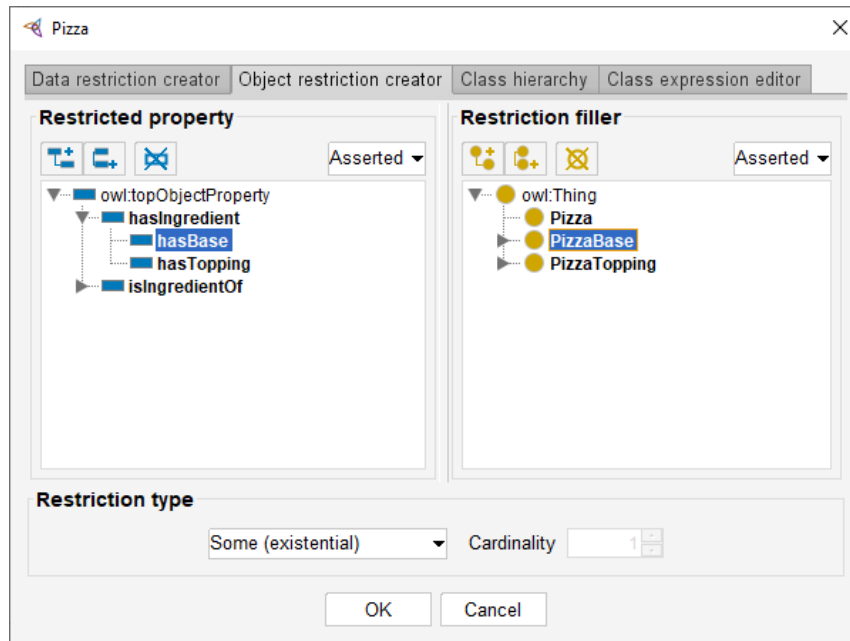


图4. 16对象限制创建器的选项卡

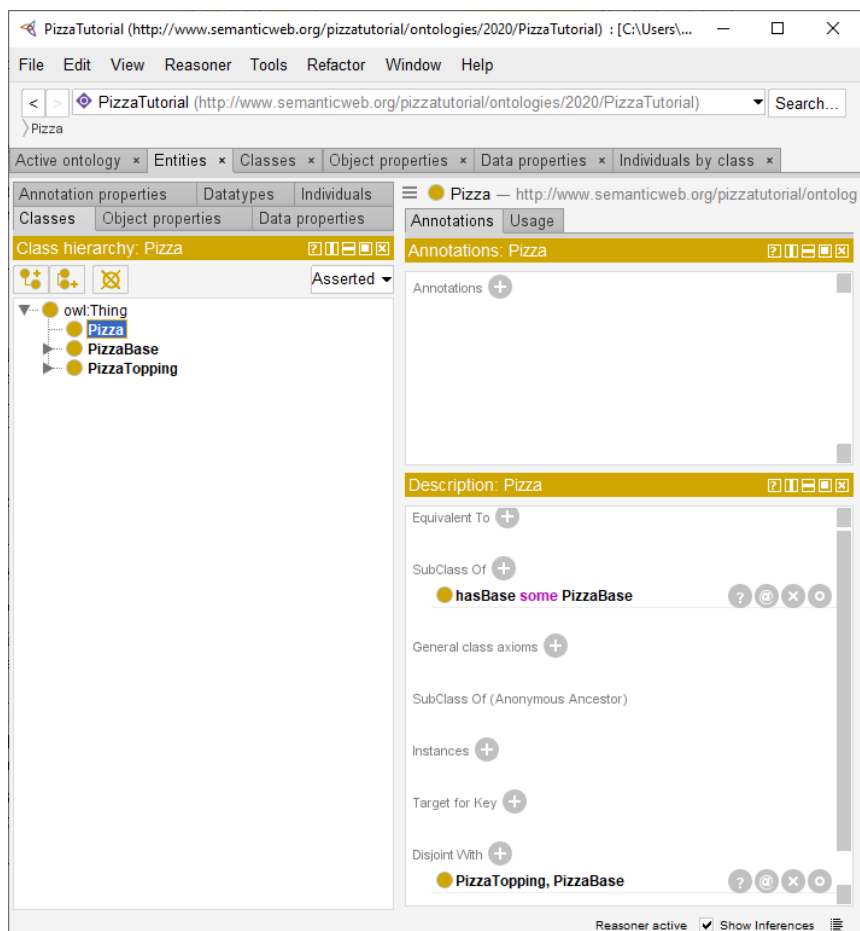


图4. 17具有hasBase限制的披萨类

我们已经把披萨类描述为事物的子类，它们有一个基础，这是某种披萨的基础。注意，这些都是必要的条件——如果某样东西是披萨，那么它就必须成为类的成员（在OWL中，所有东西都是类的成员），并且对于它有一种披萨基地是必要的。更正式地说，要成为披萨，就必须通过属性与Base的成员建立关系。

4.10.3 创建披萨的子类

现在是时候为我们的本体论添加一些不同种类的披萨了。我们会先加一个玛格丽塔披萨，这是一种含有马苏里拉奶酪和番茄配料的披萨。为了保持我们的本体整洁，我们将把不同的披萨分组为披萨。

练习14：创建披萨的子类别：命名为披萨和玛格丽塔披萨

-
1. 从“类”选项卡上的类层次结构中选择“披萨”。
 2. 单击“类”选项卡左上角的“添加子类”图标（如果您不确定，请重新查看图4.4）。您还可以将鼠标移动到图标上，您会看到每个图标的一个小弹出提示。
 3. Protege将提示您输入新子类的名称。称之为披萨。
 4. 重复步骤1-3，这次从命名披萨开始，创建NamedPizza的子类。叫它玛格丽特披萨。
 5. 使用“注释”视图向类添加注释。这是在“描述”视图的上方。补充评论：一种只有马苏里拉奶酪和番茄酱的披萨。请记住，注释属性是可以对任何实体进行断言的元数据，而对象和数据属性只能对个人进行断言。在所有的Protege本体中都包含了一些预定义的注释属性，比如注释属性。
-

在创建了这个类之后，我们现在需要指定它所拥有的配料。为了做到这一点，我们将增加两个限制，说一个玛格丽塔披萨有番茄顶和番茄顶。

练习15：创建定义标记披萨的限制

-
1. 从“类”选项卡上的类层次结构中选择标记-披萨。
 2. 单击“披萨的说明”视图中“子类别”字段旁边的“添加”图标（+）。
 3. 这再次引发了限制性对话。这次，我们将使用类表达式编辑器选项卡，而不是使用对象限制创建器。选择该选项卡。
 4. 类型可使一些莫进入该领域。而不是键入其余的名称，现在点击<控制><空格>（按住控制键，按空格键）。Protege应该为您自动完成该名称，并且该字段现在应该包含：请点击一些主题列表。这对于Protege UI的任何部分都是一种有用的技术。无论何时，只要你输入某个实体的名称，你都可以这样做

<control><space> . 如果该字符串只有一个可能的完成，那么Protege将填写适当的名称。如果有多个可能的完成情况，Protege将创建一个包含所有可能的完成情况的菜单，并允许您选择您想要的一个。

5. 单击OK以输入新的限制。

6. 重复步骤1-5，只是这次添加限制和一些断层顶部。记住使用<控件><空间>来节省键入时间。同步推理器以确保事情一致。您的UI现在应该看起来与图4. 18相似。

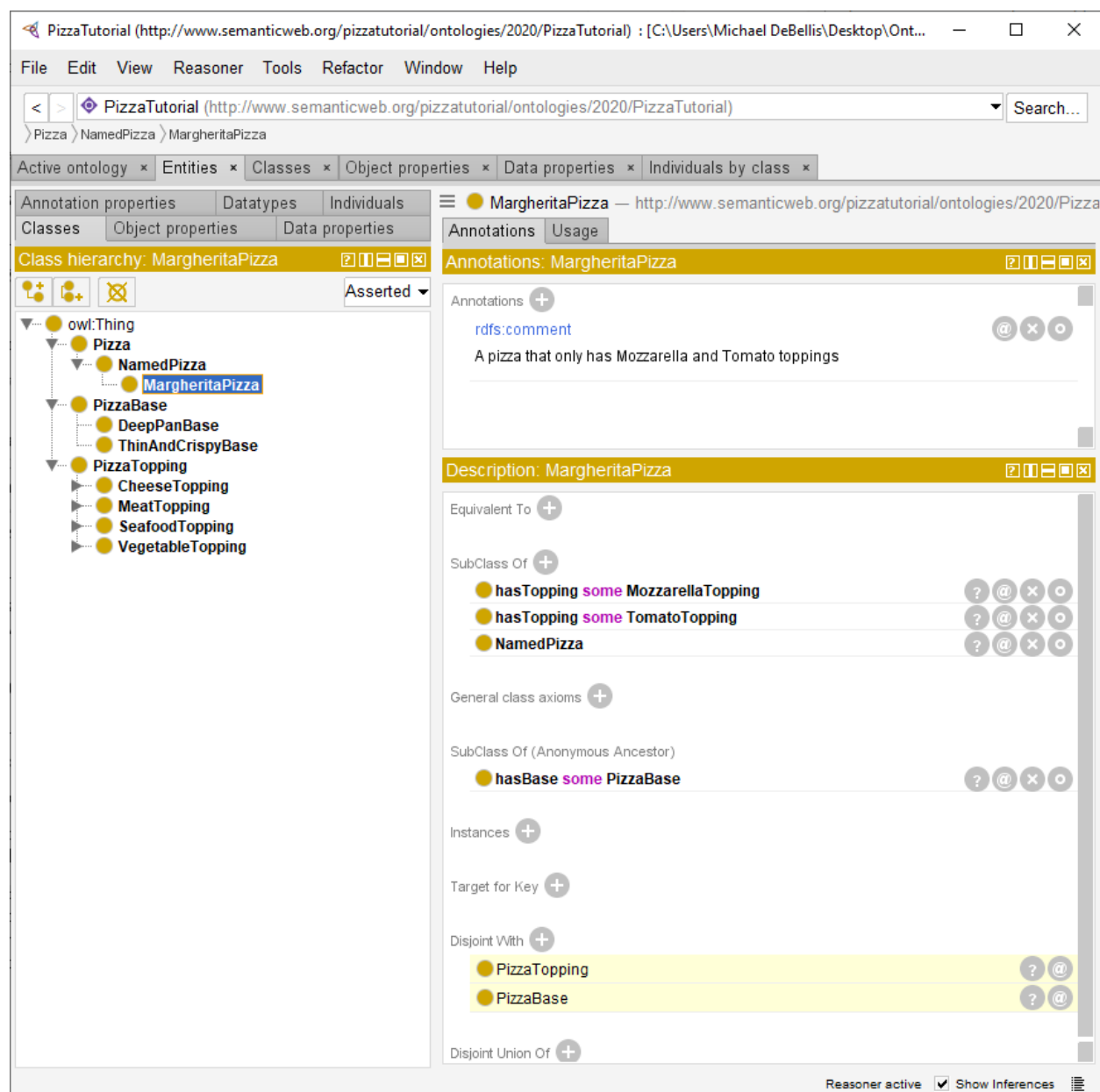


图4. 18该类的定义

注意图4.18中“不相交”下列出的两个类，并用黄色突出显示。这是从推理者中进行推理的一个例子。当我们定义披萨、披萨饼和披萨饼时，我们使这三个类不相交。也就是说，没有人可能是超过一个类的成员。由于玛格里塔披萨是披萨的一个子类，它也与披萨基地和披萨顶部不一致，所以推理者将这些信息添加到玛格里塔披萨的定义中，并与推理者的所有推论一样，用黄色突出显示了新的信息。

我们现在将创建一个班级来代表一个美国披萨，上面有意大利辣香肠、马苏里拉奶酪和番茄。因为美国披萨和披萨班类似(比如，美国披萨几乎和玛格丽塔披萨一样，但额外加了意大利辣香肠)我们将克隆玛格丽塔披萨类，然后添加一个额外的限制，说它上面有意大利辣香肠。

练习16：通过克隆玛格丽塔比萨和添加额外的限制来创建美国披萨

1. 从“类”选项卡上的类层次结构中选择标记-披萨。
 2. 选择“编辑>重复选择的类”。这将引出一个对话来复制课程。默认值是现有类的名称，因此当您启动时，将会出现一个红色的错误消息，因为您需要输入一个新的名称。把名字从美国披萨改为美国披萨。保留所有其他选项，然后选择“确定”。
 3. 确保美国披萨仍然被选中了。单击“美国披萨的描述”视图中“子类”字段旁边的“添加”图标(+)。
 4. 使用“对象限制创建者”选项卡或“类表达式编辑器”选项卡来添加附加的限制：点击一些肽键。点击。
 5. 单击OK以输入新的限制。
 6. 编辑关于美国披萨”的评论注释。它现在应该是：一种只含马苏里拉奶酪和番茄配料的披萨，因为它是从玛格里塔披萨上复制过来的。请注意，在注释的右上角有三个小图标，一个@符号，一个X和一个o。点击o。这个图标是你用来编辑Protege中任何现有数据的图标。这应该会打开一个窗口，您可以在那里编辑注释。把它改成合适的东西，比如：只有马苏里拉奶酪、番茄和意大利辣香肠酱的披萨。然后单击OK，输入对注释的编辑。
-

练习17：创建美国热比萨和太阳披萨

1. 美国比萨几乎和美国比萨一样，但上面有墨西哥胡椒。通过克隆美国披萨类，并添加一个存在限制来创建这个。

2. 苏披萨几乎和玛格丽塔披萨一样，但是有额外的橄榄和帕尔玛奶酪——通过克隆玛格丽塔披萨并添加两个存在限制，一个填充橄榄糖，另一个填充帕尔玛萨奶酪。

练习18：使命名为Pizza的子类不相交

1. 我们想让这些命名披萨的子类彼此不相交。也就是说，任何一个人最多都可以属于这些类中的一个。要做到这一点，首先选择标记披萨（或命名Pizza的任何其他子类）。

2. 单击“说明”视图底部的“不相交”旁边的(+)符号。这将显示一个类层次结构视图。使用此选项导航到命名Pizza的子类，并使用<控件><左击>选择您选择的所有其他兄弟类。然后选择“确定”。现在应该看到在NamedPizza的每个子类上显示适当的不相交公理。同步推理器。你的UI现在看起来应该类似于图4.19。

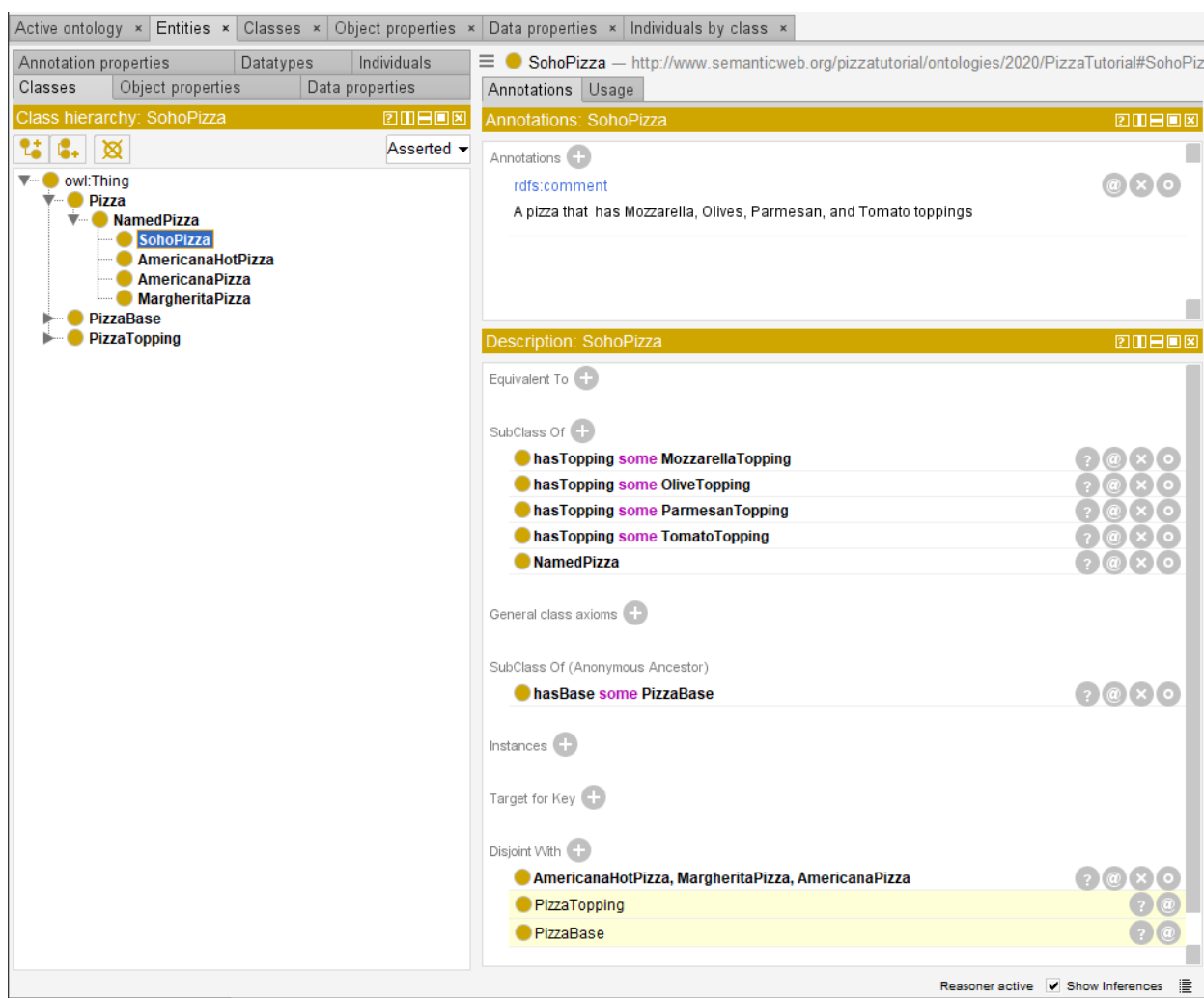


图4.19命名披萨的子类是不相交的

4.10.4 检测不能具有成员的类

接下来，我们将使用推理器来检测一个具有定义的类，这意味着它永远不能有任何成员。在当前版本的Protege中，当推理器检测到某些操作系统上的不一致或问题时，UI偶尔会锁定并难以使用。因此，为了确保您没有丢失任何工作，使用文件保存本体>保存。

有时，创建一个我们认为不可能实例化的类，以确保本体按照我们认为的那样建模是有用的。这样的类被称为探针类。

练习19：添加一个名为“命题不一致顶部”的探测类

-
1. 从类层次结构中选择具有奶酪顶部的类。
 2. 创建一个奶酪顶的子类，称为亲不一致顶。
 3. 单击“证明不一致”顶部的“描述”视图中“子类”字段旁边的“添加”图标(+)。
 4. 从弹出的对话框中选择“类层次结构”选项卡。这将显示一个小视图，看起来像您用于添加新类的类层次结构选项卡。使用此选项导航到并选择类。单击“确定”。
 5. 请确保保存您当前的本体论文件。现在运行推理器。您应该看到外观不一致顶部现在用红色突出显示，表示它不一致。
 6. 点击外观不一致的顶部，看看为什么它会用红色突出显示。请注意，在描述视图的顶部，您现在应该看到owl：等效字段下的没有。这意味着探测类等价于owl：没有。没有什么职业是猫头鹰的对立面。然而所有的个体都是猫头鹰的例子：事物，没有一个个体不可能是事物的例子
猫头鹰：没什么。猫头鹰：没有什么类等同于集合理论中的空集合。
 7. 应该有一个问题吗？就在猫头鹰右边的图标：什么都没有。与推理器的任何推理一样，可以单击新信息并对其生成解释。现在，点击这个就可以了吗？偶像这将生成一个如图4.20所示的新窗口。解释是，外观不一致顶是奶酪顶和素食顶的一个子类，但这两个类是不相交的。
 8. 单击“确定”以关闭该窗口。删除类外观不一致，选择其顶部，然后单击类视图顶部的删除类图标（见图4.4）。
 9. 同步推理器。
-

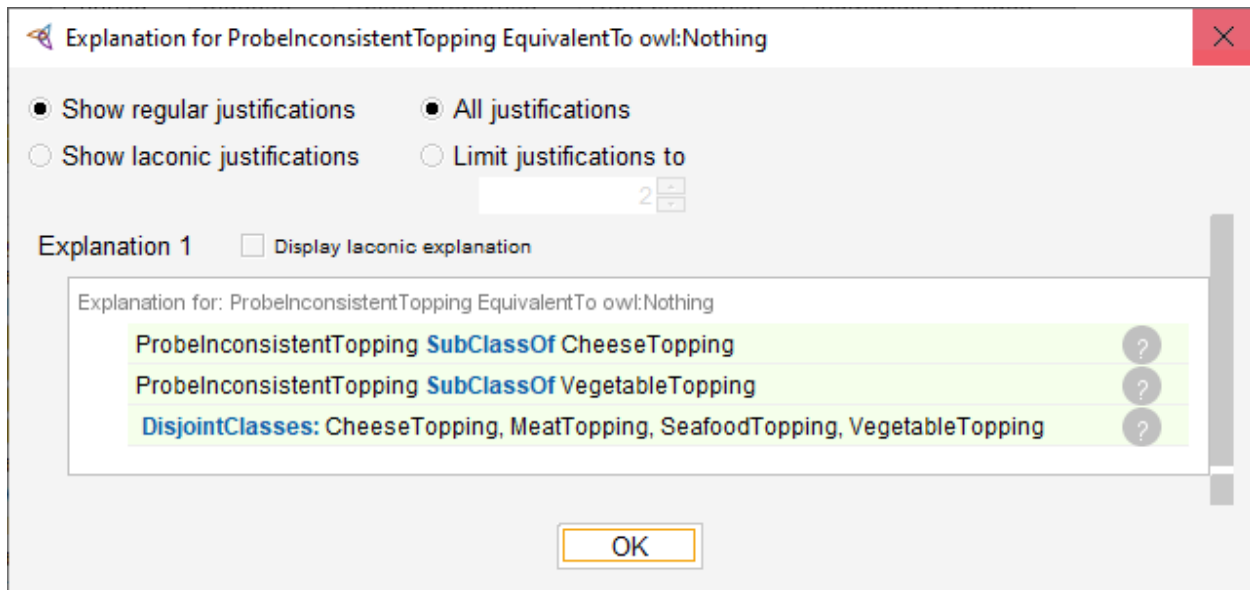


图4.20解释为什么外观不一致的顶点等同于猫头鹰：没有

4.11 原始类和定义类（必要类和充分公理）

到目前为止，我们创建的所有类都只使用了必要的公理来描述它们。必要的公理可以理解为，如果某个东西是这个类的成员，那么就有必要满足这些条件。仅凭必要的公理，我们就不能这样说：如果某个东西满足了这些条件，那么它就必须是这个类的一员。

让我们用一个例子来说明这一点。我们将创建一个披萨的子类，叫做奶酪披萨，这将是一种至少有一种奶酪配料的披萨。

练习20：创建奶酪披萨课程

1. 在“类”选项卡上的类层次结构中选择“披萨”。
2. 选择“添加子类”图标（请参见图4.4）。命名新的子类**奶酪披萨**。
3. 确保选择了奶酪披萨。单击“说明”视图中“子类”字段旁边的“添加”图标（+）。
4. 选择“类表达式编辑器”选项卡。输入新的公理：**先吃一些奶酪。请记住**，您可以使用<控件><空间>来自动完成公理中的每个单词，例如。，键入hasT，然后<控制><空间>，以自动完成其余部分。如果你没有输入足够的Protege明确地选择一个实体或描述逻辑关键字，你将会被提示与一个可能的完成数的菜单。单击“**确定**”以输入新的限制公理。



请注意，如果您只输入几个字符，可能的完成数可能会很大，从而导致一个笨拙的菜单。此外，Protege也不需要可能对可能的完成数进行类型检查等操作。例如，如果你输入“Chee”并做<控制><空间>，你会被提示使用奶酪顶部和奶酪披萨作为可能的完成，即使披萨不在><顶部的范围内。这也是推理者可以提供帮助的地方。如果您输入的类不在超出范围内，那么推理器将表示不一致。

我们目前对芝士披萨的描述是，如果某样东西是芝士披萨，那么它一定是披萨，至少要有一种奶酪配料。现在考虑一些随机的个体。假设我们知道这个人披萨班的成员。我们也知道这个人至少有一种奶酪辣酱。然而，鉴于我们目前对奶酪披萨的描述，这一知识并不足以确定个人是奶酪披萨班的成员。为了使这一点成为可能，我们需要将奶酪披萨的条件从必要条件改为必要条件和充分条件。这意味着，这不仅是成为奶酪披萨班成员的必要条件，它们也足以确定任何满足他们的随机个人必须是奶酪披萨班的成员。

一个只有必要条件的类（例如我们迄今为止定义的所有类）被称为原始类。具有充要条件的类被称为已定义的类。为了将必要条件转换为必要条件，条件必须从类描述视图中的“子类”标题下移动到“等价”标题下。这可以通过菜单选项来完成：编辑>转换为已定义的类。

练习21：将奶酪披萨从原始类转换为已定义的类

1. 确保选择了奶酪披萨。
2. 选择菜单选项：编辑>转换为已定义的类。
3. 同步推理器。

您的屏幕现在应该类似于图4.21。注意，当一个类是一个已定义的类时，它显示在UI中，它的名称旁边的圆上有三个横条纹。

到目前为止，我们已经看到推理机做了一些简单的事情，比如将不相交的公理从超类传播到子类。然而，推理者有能力做得更多。现在我们有了一个已定义的类，我们可以看到一个例子。请注意，在“类层次结构”视图中有两个选项卡。图4.21中所示的一个是断言的层次结构。这是由用户声明的公理所定义的层次结构。另一个选项卡是“类层次结构（推断）”选项卡。这是由推理者所推断出的层次结构。在我们创建一个已定义的类之前，这两个选项卡将是相同的，因为我们在本体中只有原始类。现在我们有了一个定义的类，推断的层次结构看起来会不同。选择“类层次结构（推断）”选项卡。确保推理器同步（应表示推理器激活，如图4.21）。另外，请确保在这个选项卡中扩展奶酪披萨类。您应该会看到一个类似于图4.22的屏幕。正如你应该在推断的选项卡中看到的，推理者已经推断出所有带有奶酪配料的披萨类都是奶酪披萨的子类。

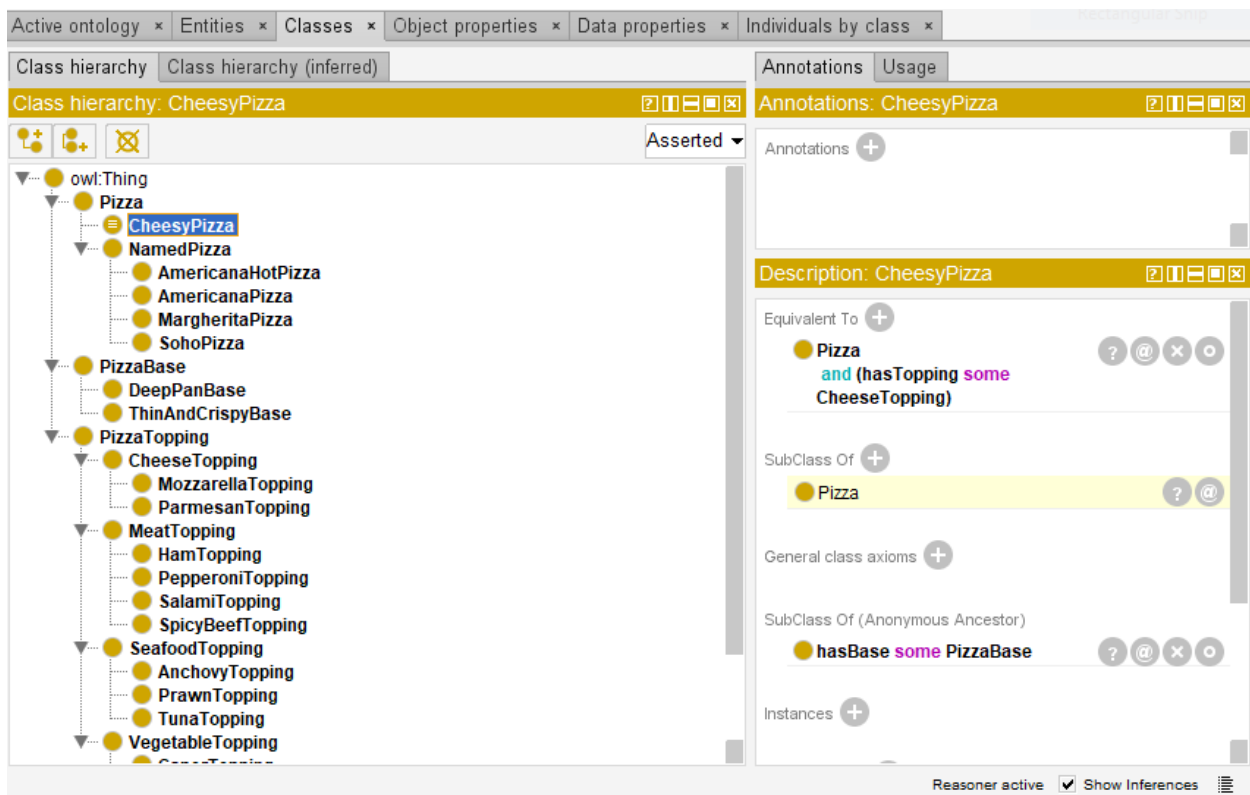


图4. 21奶酪披萨作为一个已定义类

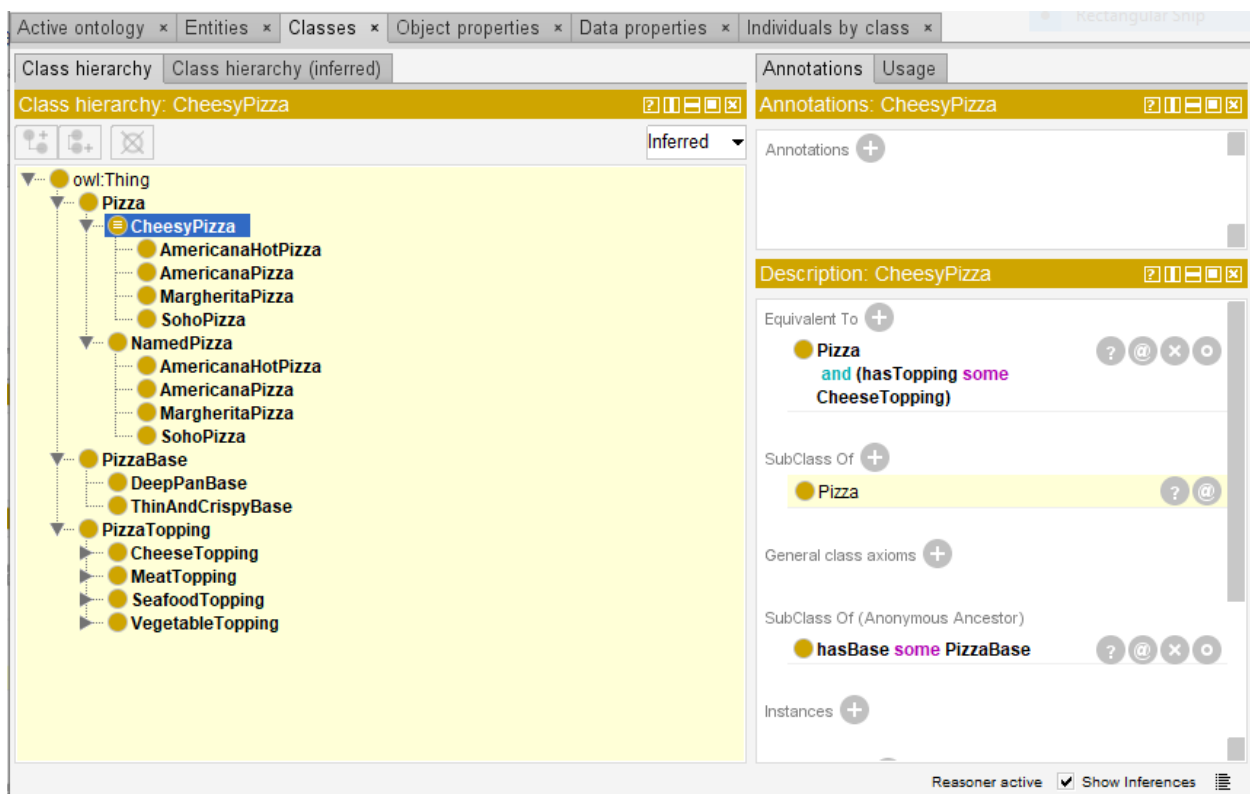


图4. 22由推理者推断出的类是奶酪披萨的子类

4.12 普遍限制

到目前为止，我们创建的所有限制都是存在性限制（使用一些DL关键字定义）。存在性限制指定在给定属性中与特定类的成员的个体存在至少一个关系。但是，存在限制并没有要求给定属性可以存在的唯一关系必须是与属于指定填充类成员的个人之间。

例如，我们可以使用一个存在的限制来描述那些至少有一种关系的个体，以及一个阶级的成员。这个限制并不意味着所有的顶部关系都必须是类的一个成员。为了将给定属性的关系限制为特定类的成员，我们必须使用通用限制。通用限制对应于一阶逻辑中的符号 \forall 。它们将给定属性的关系约束为特定类成员的个体。例如，普遍限制 \forall 描述了所有个体的关系都是植物类的成员——这些个体与不是植物类的人没有巨大的关系。

假设我们想创建一个叫做素食披萨的类。属于这类成员的人只能有一种是奶酪配料或素食配料的配料。要做到这一点，我们可以使用一个通用的限制：

练习22：创建一个定义为素食披萨的类

1. 在“类”选项卡中选择披萨饼。创建一个披萨的子类，并将其命名为素食披萨。
2. 确保选择了素食披萨。单击“说明”视图中“子类”字段旁边的“添加”图标(+)。
3. 从弹出窗口中选择“类表达式编辑器”选项卡。输入描述逻辑公理：仅添加配料（素食配料或奶酪配料）。单击“确定”。
4. 确保素食披萨仍然被选中了。运行编辑>转换为已定义类命令。
5. 素食披萨现在应该有三条水平线，就像奶酪披萨一样。此外，描述视图中的等效字段应该有：披萨和（只有顶部（奶酪配料或蔬菜配料））。请注意，创建已定义类的另一种方法是直接在等效于字段中输入描述逻辑公理。
6. 同步推理器。



这意味着，如果某样东西是素食披萨类的一员，那么它就必须是一种披萨，而且它必须只有（?通用量词）才会有一种奶酪配料或一种素食配料。换句话说，所有属于素食披萨阶层的人都必须是奶酪蛋糕或素食蛋糕阶层的成员。贝吉塔班41披萨还包括披萨，不参与任何顶级的关系。



在上面的例子等情况下，一个常见的错误是使用交集而不是联合。例如，奶酪蛋糕和素食蛋糕。虽然奶酪和蔬菜在英语中可能是很自然的说法，但从逻辑上讲，这意味着奶酪和蔬菜。这是不正确的，因为我们已经说过奶酪酱和素食酱是不相交的类，因此没有人可以是两者的例子。如果我们使用这样的定义，推理机将检测到不一致性。



在上面的例子中，人们可能会创造两个普遍的限制——一个是奶酪（?奶酪），另一个是素食（?素食）。但是，当使用多个限制（对于任何类型的限制）时，总描述被视为单个限制的交集。因此，这将相当于一个带有填充物的限制，这是番茄顶部和番茄顶部的交集——如上所述，这在逻辑上是不正确的。

4.13 自动分类和开放世界推理

确保推理机已同步（右下角的小文本应表示推理机处于激活状态）。现在，从“类层次结构”选项卡切换到“类层次结构（推断）”选项卡。你可能会注意到一些似乎令人困惑的事情。玛格丽塔披萨和苏披萨这两个班都只有蔬菜和奶酪的配料。因此，人们可能会认为，推理者会将它们归类为素食披萨的子类，因为它最近（在第4.11节中）将它们和其他食物归类为奶酪披萨的子类。这种情况没有发生的原因是一种叫做开放世界假设（OWA）的东西。这是OWL的概念之一，对于新用户甚至有经验的用户来说可能最令人困惑，因为它不同于大多数其他编程和知识表示语言中使用的近距离世界假设（CWA）。

在大多数使用CWA的语言中，我们假设目前关于该系统的所有内容都已经在数据库中了。然而，OWL原本是一种将语义带到互联网上的语言，所以语言设计者选择了OWA。开放世界的假设意味着我们不能仅仅因为它目前不在本体中就假设它不存在。互联网是一个开放的系统。这些信息可能存在于一些尚未集成到我们的本体中的数据源中。因此，我们不能得出某些信息不存在的结论，除非它被明确地表明它不存在。换句话说，因为有些东西没有被证明是真的，所以不能假定它是假的——我们假定这些知识只是没有被添加到知识库中。在我们的披萨本体论中，我们已经说过，玛格丽塔披萨有一种番茄顶，也有一种番茄顶。由于开放世界的假设，直到我们明确地说玛格丽塔披萨只有这种配料，推理者才会假设玛格丽塔披萨可能有其他配料。为了明确指定披萨有顶顶或顶顶，我们必须在顶属性上添加所谓的闭包公理。

属性上的闭包公理包含一个通用限制，即属性只能由指定的填充物填充。该限制有一个填充物，即在该财产的存在限制中出现的填充物的结合。例如，玛格丽塔比萨的顶部属性的关闭公理是一个普遍的限制，它沿着顶部属性起作用，与一个填充物是夹层顶部和断层顶的结合。也就是说，只有顶（烟雾顶或断层顶）。

练习23：在标记披萨的顶部属性上添加一个关闭公理

1. 确保在“类”选项卡中的类层次结构中选择了标记值。
 2. 单击“说明”视图中“子类”字段旁边的“添加”图标（+）。
 3. 从弹出窗口中选择“类表达式编辑器”选项卡。输入描述逻辑公理：仅顶部（顶部或虚拟顶）。
 4. 单击“确定”。
 5. 重复步骤1-4，但这次点击SohoPizzeria并使用公理：只顶（超顶或断层顶或超顶或寡顶）。
 6. 同步推理器。
-

前面的公理说，例如，任何一个玛格丽塔披萨的披萨都必须有烟雾顶和番茄顶。新的公理说，玛格丽塔披萨只能有这些配料，类似的，对于苏荷披萨和它的配料。这应该为推理者提供所需的信息，现在使他们两个的子类素食披萨。转到“类层次结构（推断）”选项卡。你现在应该看到玛格丽塔披萨和比萨都被列为素食披萨的子类。您的UI现在应该看起来类似于图4.23。请注意用黄色突出显示的各种公理。这些都是由推理者提供的额外推论。为了获得经验，你可能想要点击一些？在这些推论旁边的图标上，就可以看到推理者所产生的解释。在开发更复杂的本体时，这是一个用于调试和设计本体的强大工具。

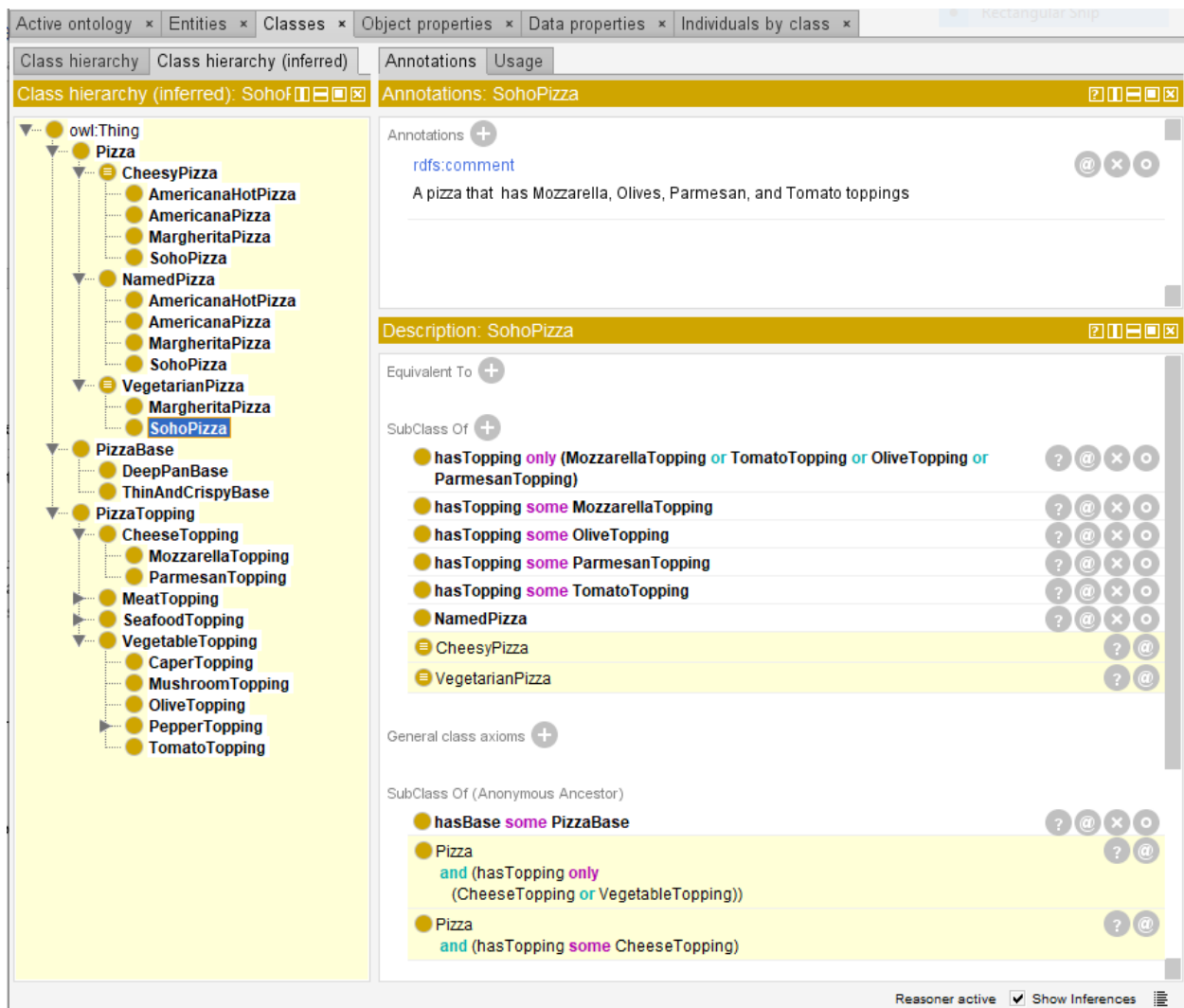


图4. 23推理者推断，玛格丽塔披萨和苏荷披萨是素食披萨的子类

4. 14 定义枚举类

面向对象编程（OOP）社区中的一个强大的工具是设计模式的概念。设计模式的思想是捕获一个处于比特定代码库更高的抽象级别的可重用模型。第一个也是最常见的设计模式之一是模型-视图-控制器模式，最初在小对话中使用，现在几乎是良好的用户界面设计的默认标准。由于OWL和标准OOP之间存在显著差异，许多关于OOP设计模式的优秀书籍并没有直接转化为OWL设计模式。此外，由于OWL的使用比OOP更近期，因此目前还没有OOP社区所拥有的关于OWL模式的优秀文档。然而，已经有许多针对OWL的设计模式，它们可以为用户提供节省时间和根据最佳实践标准化他们的设计的方法。

最常见的OWL设计模式之一是枚举类。当一个属性只有几个可能的值时，创建一个类来表示这些值，并通过列出每个可能的值来显式地定义该类是很有用的。我们将通过创建一个新的类来展示这样一个枚举类的示例

财产被称为香料，只有少数可能的值，从温和到热。在本节中，我们还将创建本体论中的第一批个体。

练习24：创建一个枚举的类来表示披萨的味道

1. 创建一个新的猫头鹰子类：一种叫做香料的东西。
2. 确保已选择了“辣度”。单击“说明”视图中的“实例”字段旁边的“添加”图标(+)。
3. 系统将提示您出现一个如图4.24所示的窗口。顶部的钻石图标是用于创建一个新的个体。带有X通的圆圈是用来删除一个个体的。使用菱形图标创建3个个体：热度、中度和温和度，所以你的UI看起来像图4.24，然后点击OK。
4. 你可能会注意到，只有一个新个体是作为香料的实例创建的。没关系。下一步将为推理机提供足够的信息，使其他两个也是香料的实例。
5. 确保香料仍然被选中。单击“说明”视图中“等效对象”字段旁边的“添加”图标(+)。这次，我们将通过直接为其输入类的定义到这个字段中来创建一个已定义的类。选择“类”表达式编辑器选项卡，然后输入DL公理：{热、中等、温和}。选择“确定”。
6. 现在运行推理器。您应该会看到，香料现在是一个已定义的类，而所有三个人：热、中和温和，现在都是这个类的例子。

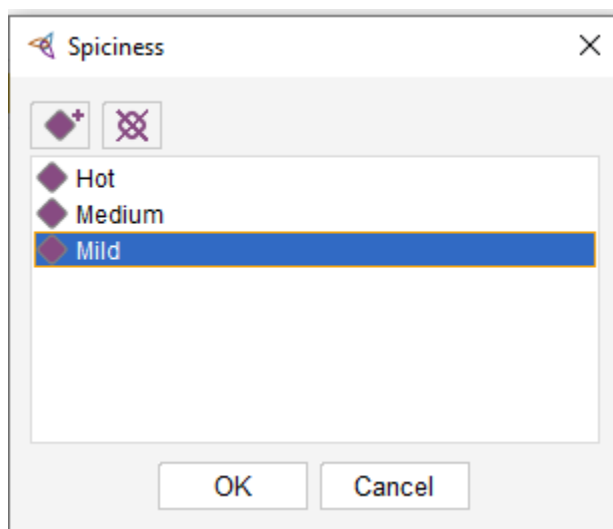


图4.24为一个枚举的类创建个人

4.15 添加香料作为一个属性

接下来，我们需要添加一个属性来定义辣酱酱的辣度。

练习25：创建和使用“快速”属性

1. 进入“对象属性”选项卡。创建一个名为快速特性的新属性。将其域定义为香料，将其范围定义为香料。运行推理器，以便它了解新的属性。
2. 返回“类”选项卡，选择类。单击“子类”字段旁边的“添加”图标(+)。输入DL公理：速度值热。请记住，您可以使用<控件><空间>来自动完成。单击“确定”。
3. 请注意，这是一种与以前不同的限制类型。在我们定义抽象的限制之前，比如一些。也就是说，来自一个类的一些值，但是特定的个体没有被指定，只要它是来自那个类的个体，限制就得到满足。现在，我们正在定义一个与特定个体相关的限制，因此，我们使用value关键字，而不是使用一些或仅使用的关键字。
4. 现在，我们将使用这个属性来定义一个新的披萨类。首先创建一个新的披萨子类，叫做香料披萨。
5. 确保选择了香料披萨。单击“子类”字段旁边的“添加”图标(+)。输入DL公理：超出一些(香料值热)。这就是说，香料披萨一定有辣的味道。
6. 通过选择并使用编辑>转换为定义的类转换为定义的类。运行推理器。

现在转到“类”选项卡中的类层次结构(推断)选项卡(参见图4.25)。你应该会看到，美国辣披萨现在被归为香料披萨的一个子类，因为它有一个辣的配料(辣椒酱)的辣值。

4.16 基数限制

在OWL中，我们可以描述与其他个体或最多数据类型值至少或指定数量的关系的个体类别。描述这些类的限制被称为基数限制。对于给定的属性P，最小基数限制指定了个体必须参与的P关系的最小数量。最大基数限制指定了个人可以参与的P关系的最大数量。基数限制指定了个人必须参与的P关系的确切数量。关系(例如两个个体之间的关系)只有在可以确定填充关系的个体是不同的情况下才被视为单独的关系。

让我们在我们的披萨本体中添加一个基数限制。我们将创建一个新的披萨子类，称为有趣的披萨，它将被定义为有3个或更多的配料。

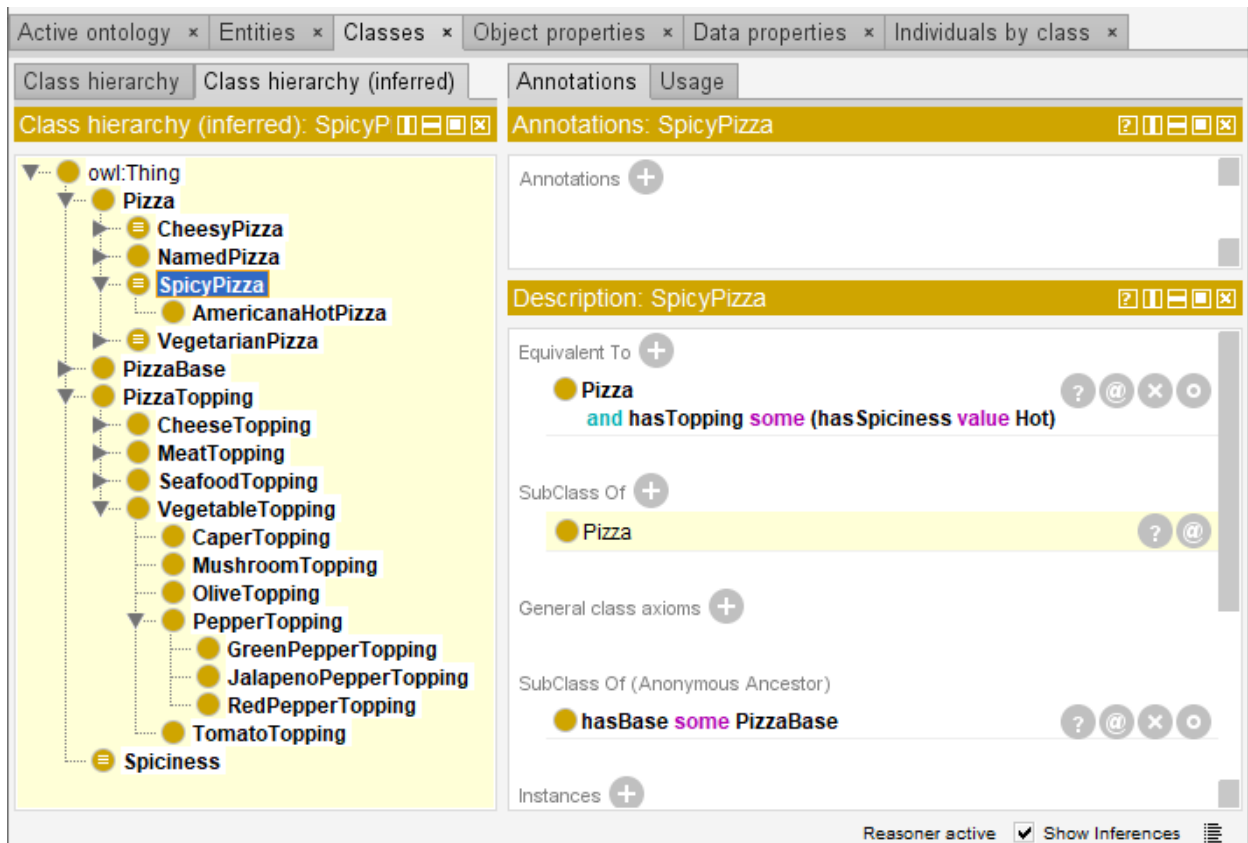


图4. 25美国热披萨被分类为香料披萨

练习26：创建一个至少有三个配料的有趣的披萨

1. 创建一个披萨的**子类**，叫做**有趣的披萨**。
2. 单击“子类”字段旁边的“添加”图标（+）。使用类表达式编辑器选项卡，输入点击min 3 点击，然后单击确定。
3. 确保**有趣披萨**仍然被选中，并使用“编辑>转换为已定义的类”选项将有趣披萨转换为已定义的类。
4. 运行推理器。

转到“类别”选项卡中的“类别层次（推断）”选项卡，然后单击“兴趣萨”。你应该会看到有三个披萨类被归类为有趣的：**美国热披萨**，**美国披萨**，和**苏霍披萨**。

第五章数据类型属性

到目前为止，我们一直在描述对象的属性。这些属性是有一个类的范围。与大多数其他面向对象的语言一样，OWL还具有定义具有简单数据类型范围的特性的能力，如字符串或整数。对象纯粹主义者会认为所有的东西都应该是一个对象。然而，借用《神奇蜘蛛侠》中的一句话：“巨大的力量会带来巨大的开销。”也就是说，类和实例的额外功能也意味着实例占用更多的空间，处理速度可能比简单的数据类型更慢。因此，OWL附带了一个大量的数据类型库，这些数据类型大多是从XML导入的。这就是为什么Protege中的许多预定义数据类型都有一个xsd的前缀，例如xsd：字符串和xsd：整数。还可以创建新的基本数据类型。但是，对于大多数用例，如果需要一个不映射到预定义类型之一的数据类型，那么最好的解决方案通常是只定义一个类。

具有简单数据类型范围的属性称为数据类型属性。这类似于统一建模语言（UML）OOP建模语言中的关联和属性之间的区别。UML关联类似于OWL对象属性，而UML属性则类似于OWL数据类型属性。它也类似于实体关系建模中的关系和属性之间的区别。E/R模型中的关系类似于OWL中的对象属性，而属性也类似于数据类型属性。因为数据类型没有OWL对象的所有功能，所以4.8节中描述的许多对象属性功能，例如数据类型属性。

5.1 定义数据属性

与其他OWL实体一样，数据类型属性可以通过“**实体**”选项卡中的“**数据属性**”选项卡或“**数据属性**”选项卡来定义。

我们将使用数据类型属性来描述披萨的卡路里含量。然后，我们将使用一些数字范围来大致分类特定的披萨为高热量或低卡路里。为了做到这一点，我们需要完成以下步骤：

1. 创建一个数据类型属性，它将用于声明特定比萨的卡路里含量。
2. 创建几个具有特定卡路里含量的披萨人的例子。
3. 创建两类，大致将披萨分为低热量或高卡路里。

练习27：创建一个名为数据内容的数据类型属性

-
1. 打开“**数据属性**”选项卡。选择owl：topData属性。
 2. 单击左上角中的“添加子属性”图标。这与添加对象属性的UI完全相同。
 3. 将新的数据属性命名为**计算内容**，然后选择确定。
 4. 单击“**域**”视图中的（+）图标。使用“**类层次结构**”选项卡可以选择Pizza类别作为域。

5. 单击“描述”视图中的“范围”旁边的(+)图标。从弹出菜单中选择内置的数据类型选项卡。从可能的内置数据类型的相当长的菜单中选择xsd:整数⁷。这是用于整数数据属性的默认数据类型。
6. 单击“描述”视图旁边的“功能”复选框。一个披萨只能有一种高热量的含量，因此是功能性的。数据属性通常是功能性的。
5. 选择OK并运行推理机。您的UI应该看起来与图5.1相似。

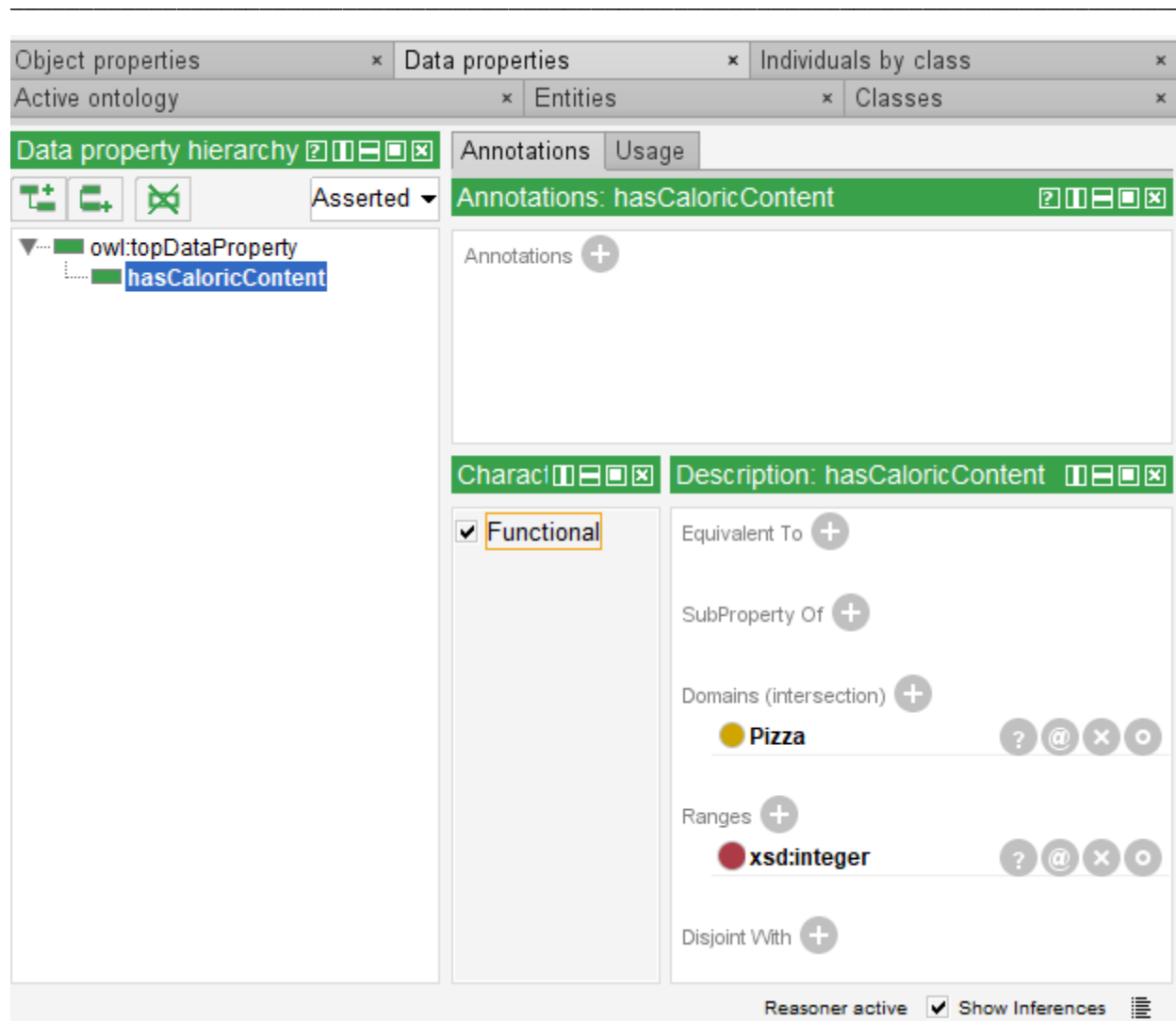


图5.1数据数据属性

⁷ 由于历史原因，有许多数据类型很少被使用，例如，xsd: int，它类似于xsd:整数。对于数字，默认的数据类型是xsd:整数表示整数，xsd:十进制表示实数。除非您有很好的理由来使用不同的数字数据类型，否则最好坚持使用这些默认类型。例如，当你写一个SWRL规则时，如果你使用一个数字，SWRL将推断任何整数都是xsd:整数，任何基本原理都是xsd:十进制。

请注意，与对象属性一样，定义域和/或范围是可选的。通常，这样做是一种很好的做法，因为它可能导致在建模阶段而不是在运行时发现本体中的错误。

5.2 定制原型机用户界面

为了演示我们的新数据属性，我们将需要创建一些Pizza类的实例，并设置数据属性的值。Protege的优点之一是，它可以高度定制到您的特定需求和工作风格。有许多可用的视图没有包含在默认的Protege环境中，因为它会太杂乱。此外，可以调整已经使用过的所有视图的大小、删除或添加到现有选项卡中。您也可以创建自己的全新选项卡。

作为一个例子，我们将首先提出一个新的主要标签，称为**个人的分类**。此选项卡可用于创建个体以及添加或编辑其对象和数据属性值。我们将自定义这个选项卡，通过添加一个新视图使其更容易使用。

开始使用菜单选项窗口**>选项卡>个人按类打开这个**新选项卡。当然，如果它已经存在于你的UI中，只需选择它。

我们想添加一个新的视图，并在该视图中作为一个额外的子选项卡，**该视图目前有注释和用法**，这些选项卡靠近右上角⁸。一旦你进入**个人按类选项卡选择窗口>视图>个人视图>个人按类型（推断）**。这将给您一个新视图的蓝色轮廓。当您围绕现有窗口移动轮廓时，它将根据您的移动方式发生变化，并指示在单击后它将如何适应现有选项卡。当蓝色轮廓看起来像图5.2时，单击左侧，您将看到新视图添加为另一个子选项卡。

在你点击后，你的UI现在应该看起来类似于图5.3。如果你点击了其他地方，你可以进入新视图，通过**点击**视图右上角的X删除它，然后重新做并正确定位它。一开始，它看起来有点不直观，但在你做了几次之后，就很容易定位新视图。

使用这个新视图，您可以看到类下面显示的每个类的实例。每个类都可以进行扩展或收缩，以查看或隐藏其特定的实例。由于我们的本体中没有很多实例，但是这个新视图的有用性并不是那么明显，但是当我们添加更多实例和将来处理更大的真实本体时，这个视图对于找到类的特定实例非常有用。注意，UI只是显示个体实例的最直接的类（或类）。例如，我们目前只有三个人，香料的三个例子：热度，中度和温和度。这些也是猫头鹰的实例：Thing（和所有实例一样），但是UI只将它们显示为香料的实例，因为它暗示它们也是所有香料的超类的实例。

⁸ 您的特定的Protege UI可能看起来与一些屏幕快照略有不同，这取决于您的组织或其他用户是否已经定制了Protege UI。如果您曾经想要返回一个主选项卡的默认配置，选择该选项卡，并使用**窗口>重置选定选项卡的默认配置**。

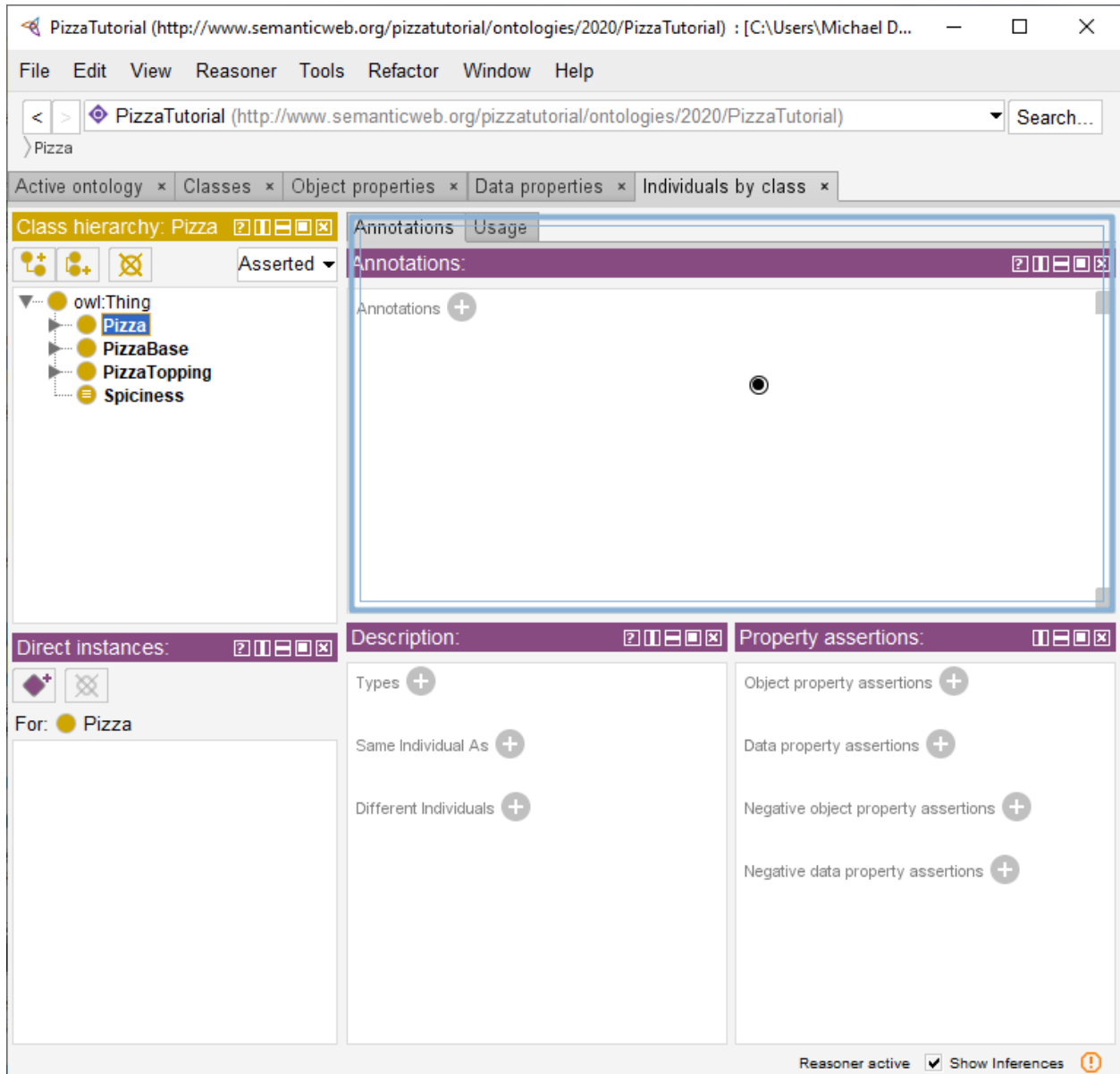


图5. 2将新视图添加到按类划分的个人视图选项卡

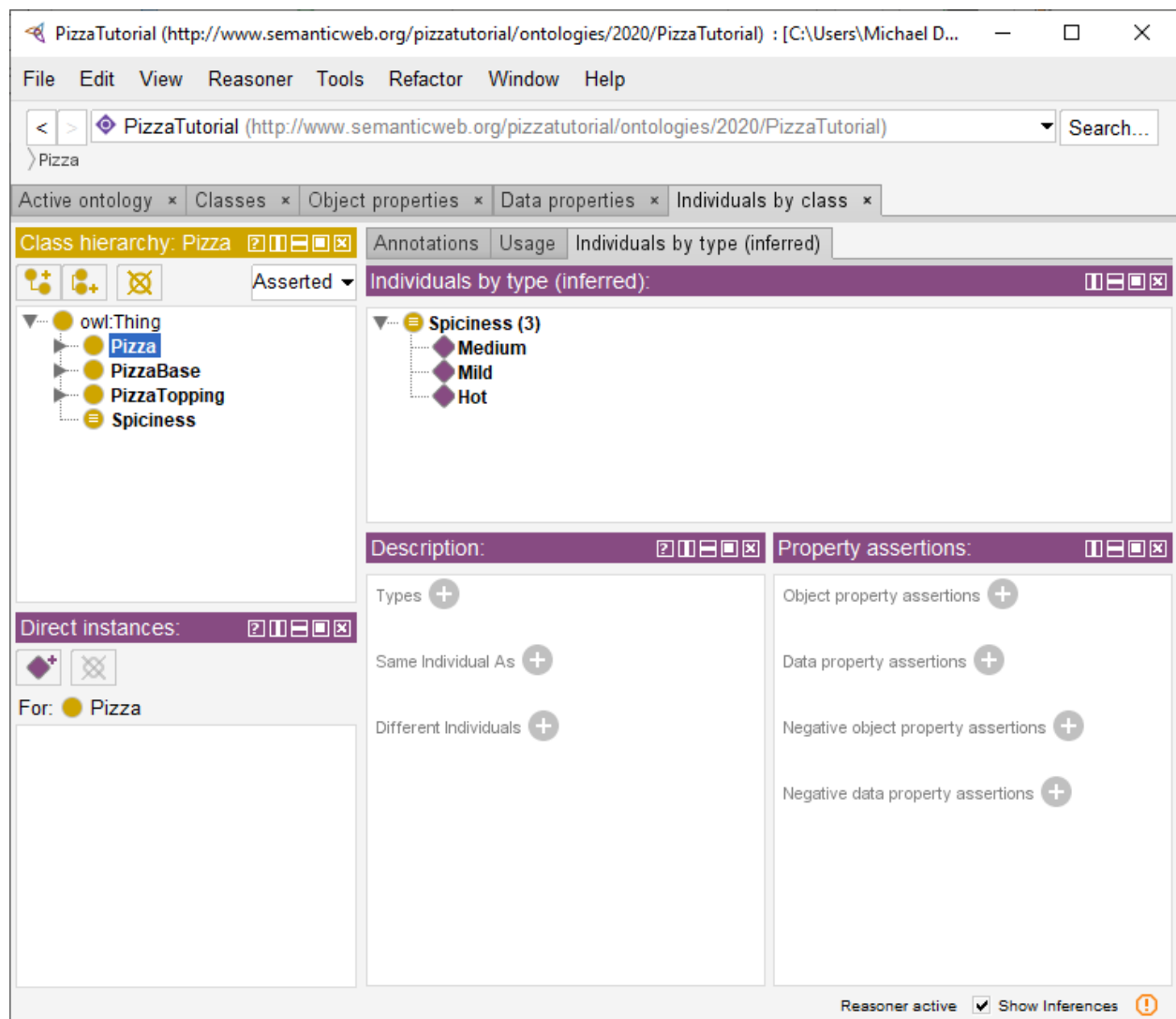


图5.3 A “按类划分的自定义个人”选项卡

练习28：创建披萨个人的例子

1. 现在我们将添加我们的第一个真正的披萨。保留在“**个人按类**”选项卡中。
2. 使用**左上角**的**类层次结构**视图导航到标记**披萨**并**选择**它。在类层次结构选项卡下有一个**直接称为直接实例的视图**。点击那个视图中的小钻石。这将提示您输入您的新个人的名字。叫它**玛格丽特披萨1**。
3. 您的UI现在应该看起来与图5.4相似。

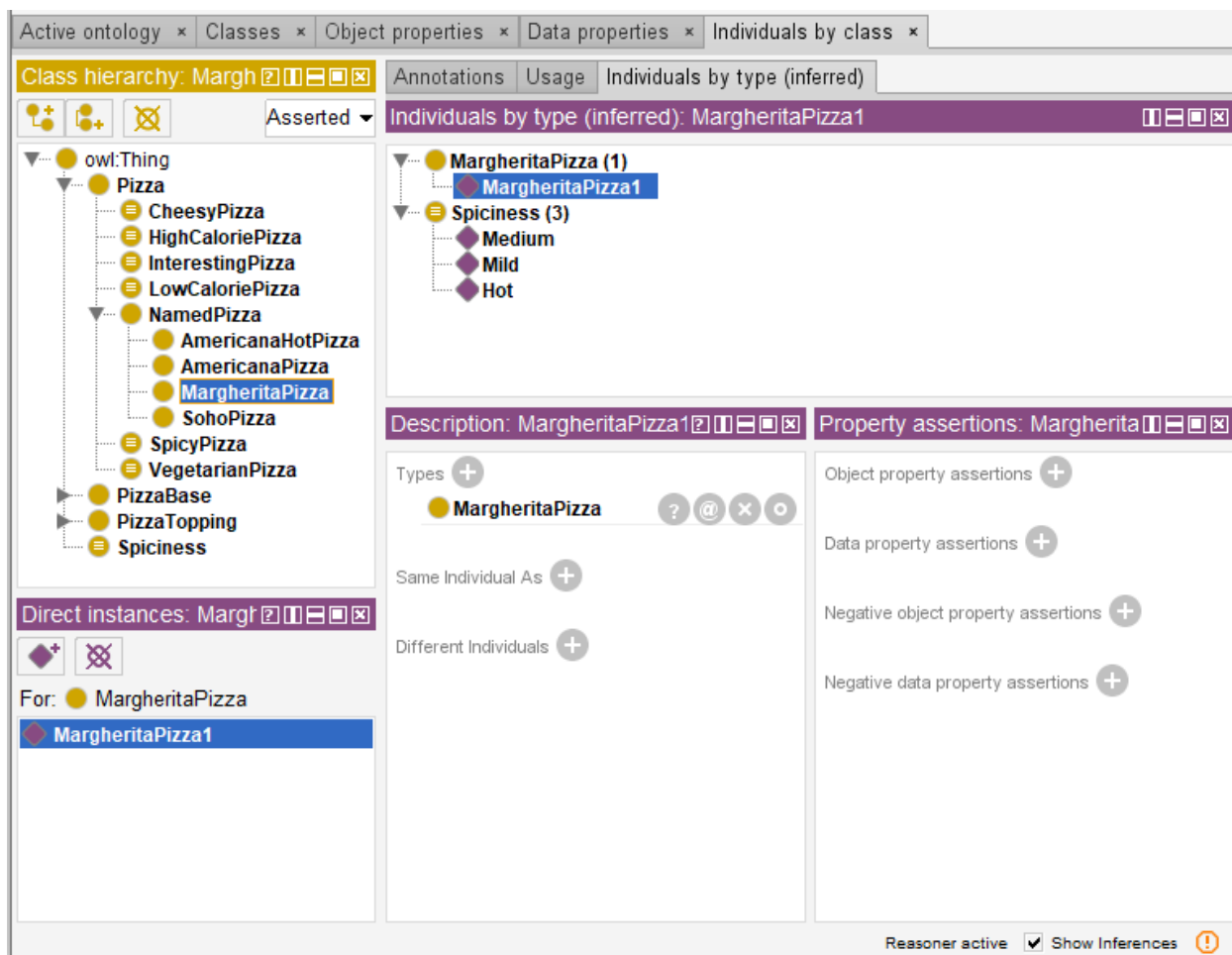


图5. 4创建了我们的第一个披萨

练习29：分配一个数据属性值

1. 保留在“**个人按类**”选项卡中。点击**玛格丽塔披萨1**。您应该在描述视图中看到，它是玛格丽塔披萨的一个实例。现在，您将使用**属性断言**视图来设置标记披萨1的热量内容。此视图可用于设置对象和数据属性。
2. 单击右下角的“**属性断言**”视图中的**数据属性断言**旁边的(+)图标。
3. 使用弹出窗口选择数据属性的**热量内容**。然后输入263作为值，并使用底部的菜单将该值的数据类型定义为xsd:整数。**注意：这与您**在练习27中所做的有所不同。在练习27中，您已经为该属性定义了数据类型。在这里，您将为一个特定的值定义数据类型。如果Protege可以只为您推断出数据类型，那就太好了，但是因为数据类型定义可能很复杂，这比看起来更难，所以您需要确保为每个值显式地定义数据类型。当您进入更现实的本体时，您将经常使用像Cellfie（见第8章）这样的工具来自动加载您的个人数据，这些工具可以自动为每个个体添加数据类型信息，作为加载过程的一部分。

4. 您的UI现在应该看起来与图5.5相似。选择“**确定**”以输入新的值。运行推理器。

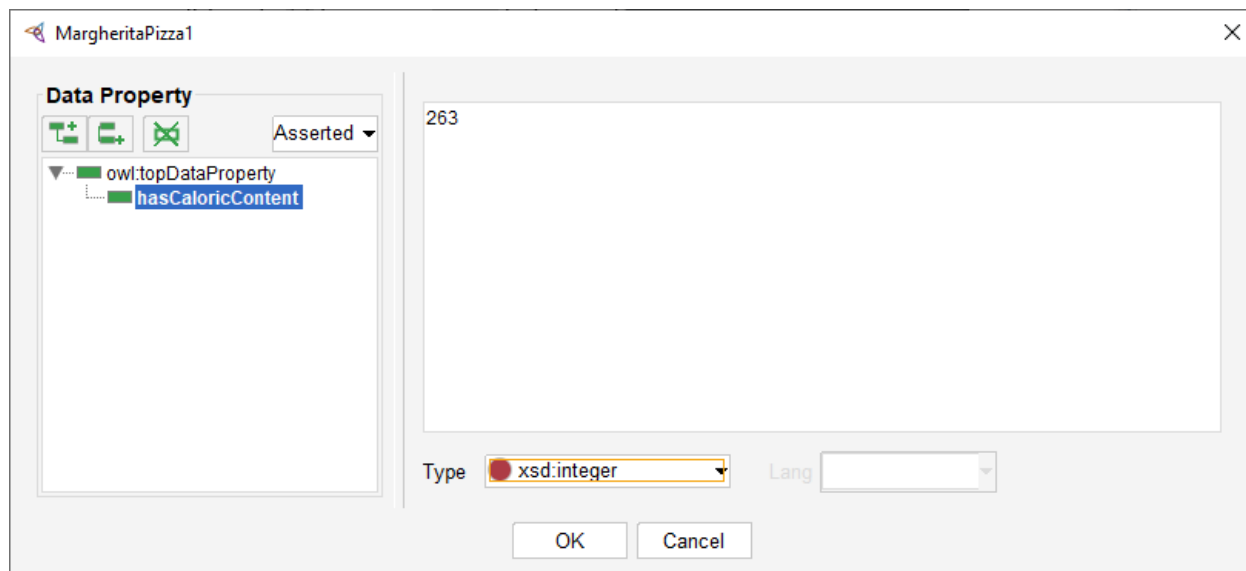


图5.5 示披萨的热量含量1



本体论中最常见的错误来源之一是对数据属性值有错误的数据类型。您越早发现这些错误，就越容易调试它们，因此在输入任何值后经常运行推理器是一个好主意。请注意，在Protege 5.5的某些版本中。存在一个小错误，UI可能由于数据值不一致（例如，整数属性中的字符串值）而锁定。如果发生这种情况，最好的做法是尽可能保存您的工作，退出Protege，然后重新启动它。当您重新启动时，它将在运行推理器之前修复数据类型错误，然后运行推理器以确保您确实已经修复了该错误。

练习30：创建更多的实例和数据属性值

1. 保留在“**个人按类**”选项卡中。点击其他披萨并创建它们的实例(apx. 5-10)，然后填写其值在200到800之间的热量含量。试着让你有一半的披萨超过400卡路里，一半不到400卡路里。UI保留了以前使用的数据类型，所以一旦您定义了第一个热量内容，您就不需要再次设置数据类型，但最好确保它是正确的，在这种情况下：xsd：整数。

2. 对于您的实例，坚持一个直观的命名标准是一个好主意，比如<类名><Number>，就像我们对标记披萨1所做的那样。根据你实例化的类，你的披萨应该有像玛格丽塔披萨2，Soho披萨1，等等。

3. 一定要创建一个叫做美国披萨1的美国披萨的实例，其热量含量为723。
 4. 请确保在创建所有实例后运行推理器。
-

练习31：创建每个比萨都包含热量内容的数据类型限制

1. 导航到“类”主标签。
 2. 选择披萨班。
 - 3 单击描述视图中“子类”字段旁边的(+)图标。这次让我们使用数据限制选项卡。在“受限制”属性视图中导航到并选择“计算内容”。在“限制填充”视图中，向下滚动到xsd:整数，并选择它。限制类型应该设置为默认值，这是一些。如果它不使用菜单来改变它。你的UI应该如图5.6所示。单击“确定”。
 4. 注意，您也可以选择精确1，因为披萨只能有一个热量内容，但由于您已经定义了属性为功能，这是不必要的，一些或精确1具有相同的效果。正如Protege通常提供几种方法在用户界面中输入相同的信息一样，OWL通常提供不同的方法在模型中提供相同的信息。最好的事情是，推理机让你不用太担心你的操作方式，只要你的定义是一致的。
-

我们现在已经声明，每个披萨都有热量含量，而且内容必须是一个整数。除了使用预定义的数据类型集之外，我们还可以通过指定对可能的值的限制来进一步专门化数据类型的使用。例如，很容易为一个值范围。

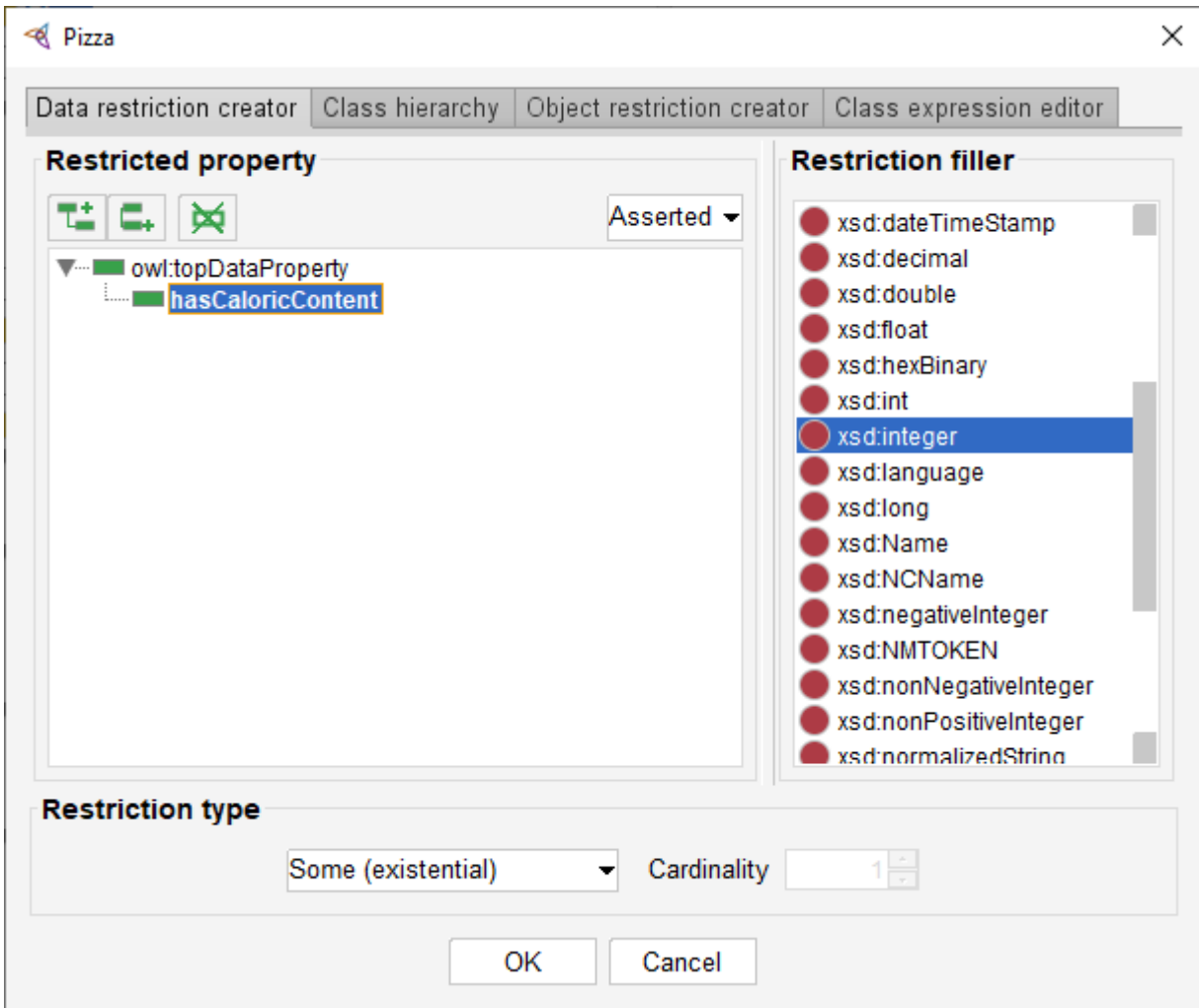


图5.6定义数据数据属性限制

使用我们创建的数据类型属性，现在将创建定义类，指定有趣值的范围。我们将定义高热量披萨为任何热值等于或高于400的披萨。

练习32：创建一个高卡路里披萨定义的类

1. 导航到“类”选项卡。
2. 选择披萨。创建一个披萨的子类，称为高卡路里披萨。
- 3 确保选择了高热量的披萨。单击描述视图中“子类”字段旁边的（+）图标。在类表达式编辑器类型中，已包含一些xsd：整数[>=400]，然后单击“确定”。
4. 确保高热量披萨仍然被选中，并使用编辑>转换到已定义的类，使其成为一个已定义的类。

5. 重复步骤1-4，但这一次创建一个披萨子类，使其定义为：内容一些xsd:整数[<400]。
6. 运行推理器。你现在应该看到，每个热量含量大于或等于400的披萨都被归为高卡路里披萨，同样，那些热量含量低于400的披萨也被归为低卡路里披萨。请参见图5.7中的“描述”视图。

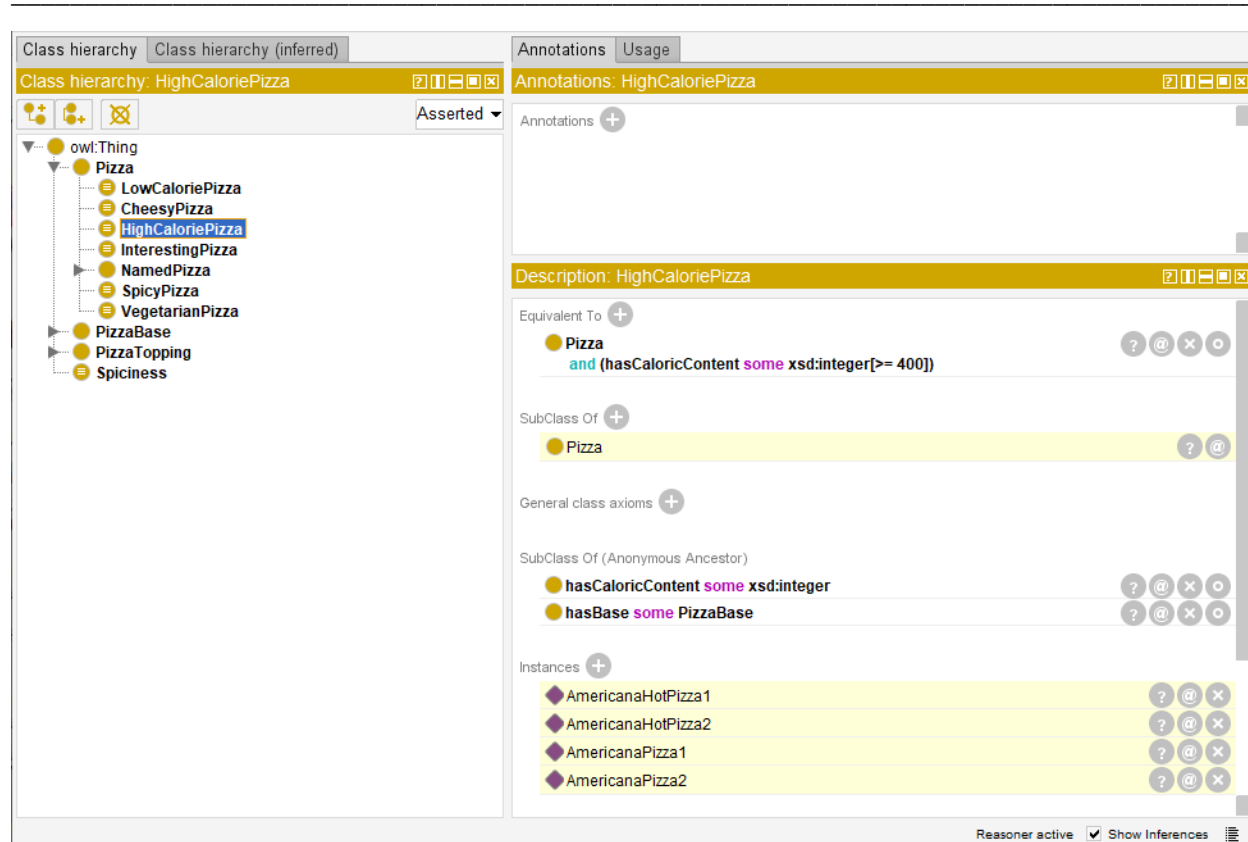


图5.7高热量披萨

第6章为已枚举的类添加顺序

在本章中，我们将扩展我们在第4.14章中创建的用来建模辛辣度的枚举类。本章将重点介绍OWL中对象属性的一些功能。我们将为香料的实例创建一个订单。也就是说，热比温和，辣比温和。若要开始，请进入“对象属性”选项卡。创建一个作为的子属性的新属性 `owl:top` 对象属性。称此属性为香料。使其域和范围为香料类。使属性具有传递性。传递意味着如果X大于Y，Y大于Z，那么X大于Z。这当然类似于数学中大于和小于关系的关系。创建另一个名为 `MilderThan` 的属性。使一个属性与另一个属性相反。不管是哪一个，你只需要指定一个属性是另一个属性的逆，推理者就会意识到两者都是逆的。运行推理器。你会看到推理者已经推断出了氨基酸的域和范围，以及它是传递的和香料的逆。

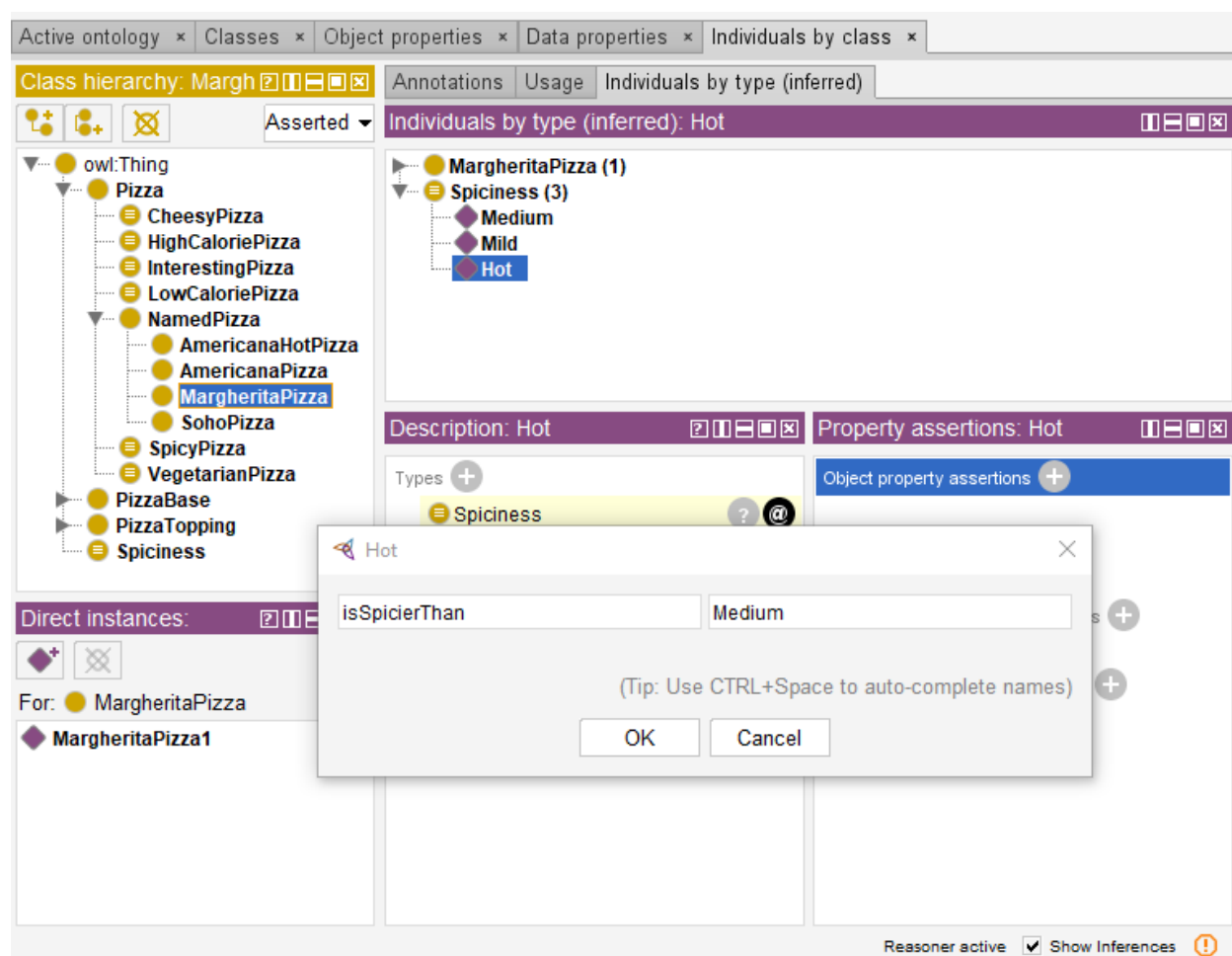


图6. 1在“按类划分的个人”选项卡中设置“按比例大于比例”的属性

接下来返回到个人的类标签。转到“按类型（推断的）划分的个人”视图。你应该看看现在存在的个体。到目前为止，我们有您创建的披萨和香料的例子：热的、中号的和温和的。单击“热”。请注意，在右下角的属性断言视图中，标题现在应该说：属性断言：Hot。单击对象属性断言旁边的（+）图标。系统将提示您使用一个包含两个区域的表单来输入值。该属性的名称位于左侧，其值位于右侧（请参见图6.1）。输入香料比作为属性的名称。请记住，您可以使用自动完成，所以您应该只需要输入isS和输入<控件><空间>和Protege将填写属性的名称。输入介质作为值。您的UI应该看起来与图6.1相似。选择“确定”。现在单击“中”并将其“大小大小”值设置为“温和”。这就是您需要做的所有数据输入工作。现在再次运行推理机，并点击热、中、和温和的个人。您应该看到，所有附加的其他值和其他值都已为您填充，因为推理者知道这两个属性是逆的和可传递的。例如，我们根本没有编辑的温和，应该有两个值。

我们可以用各种方式使用这些属性来推断事物的相对辣度。我们将在第8章中展示一些例子。



这就总结了使用Protege设计类和属性的基础知识。还有一个Protege的[athttps://webprotege.stanford.edu/#This](https://webprotege.stanford.edu/#This)可以带你到一个页面，你可以通过提供电子邮件和创建密码来创建一个帐户。Web Protege支持多个用户，并具有额外的功能，例如用于协作开发的线程讨论。但是，它目前不支持任何推理机，所以将WebProtege中开发的本体引入桌面版本来运行推理机并验证本体是一个好主意。有关Web程序学的更多信息，请参阅第12章。

第七章名称：IRI 的名称、标签和名称空间

在练习2中，我们设置了一些关于新的实体名称和渲染的参数，而没有进行太多的解释。在OWL中使用名称的一个概念有点复杂，所以我们想等到您对本体有了基本的了解之后，然后再深入研究这些细节。

首先，请记住，本体中的每个实体都有一个唯一的国际化资源标识符（IRI）⁹。IRI类似于URL。事实上，URL是一种IRI。也就是说，所有的url都是iri，但许多iri不是url。URL通常用来标识要在浏览器中查看的特定页面。IRI通常粒度较小，对于任何类型的资源，而不仅仅是那些意味着要在浏览器中查看的**资源**。如果您转到Protege中的**活动本体**选项卡，您将看到本体的本体IRI。这是所有实体共同具有的基本IRI。此外，每个实体都有一个在基础IRI之后唯一标识实体的IRI。

您可以通过单击任何实体并启动（但不完成）**重构>重命名实体**命令来看到这一点。点击披萨课程。然后选择**重构>重命名实体**。你将得到一个弹出窗口，当前名称：披萨。然而，这只是IRI的最后一部分。要查看完整的IRI，请单击右下角的复选框：**显示完整的IRI**。你的完整IRI将有所不同，但它看起来会有些like：<http://www.semanticweb.org/pizzatutorial/ontologies/2020/PizzaTutorial#Pizza>。取消选中“**显示完整IRI**”框，然后取消“**重命名**”命令。

如果您回顾练习2，在创建新实体时有两个选项。一是使用用户提供的名称。这是您应该在教程开始时选择的选项，并且现在应该是活动的。另一种方法是使用一个自动生成的名称。此选项将为每个实体的IRI创建一个通用唯一标识符（UUID）。UUID是一个由算法生成的ID，并保证它是唯一的。还有两种显示实体的方法。一种方法是使用IRI的最后一部分，通常出现在#符号后面，如上面的披萨例子。另一种方法是使用一个称为标签的注释属性。注释属性旨在提供有关实体的元数据。对象和数据属性值只能在个人上断言。但是，由于所有实体都具有元数据注释属性，因此可以断言到任何实体。任何Protege本体都默认包含一些注释属性。您可以通过查看“**注释属性**”选项卡来查看**这些内容**。**请注意**，与其他属性一样，您也可以添加自己的注释属性，但它们应该用于元数据，而不是常规数据。您将看到rdfs:label是默认的注释属性之一。当您对实体Iri使用uuid时，默认情况下，Protege将自动使用您在rdfs:标签注释属性中为新实体输入的名称。虽然您也可以将Protege配置为可以使用其他属性，但也可以使用与练习2中使用的相同的实体呈现对话框。

这两种选项都有优点和缺点，两者之间也有选项，比如使用用户为iri提供的名称和使用rdfs:标签来获得更直观的名称。细节可能会变得复杂，而且在社区内对于哪个通常更好也没有普遍的共识。对于您的第一个本体，由于您将使用SPARQL，我选择使用用户提供的实体名称，因为它是更简单的选项，特别是对于SPARQL查询，您将在中看到

⁹iri有时也被称为uri。URI与IRI相同，只是URI只支持ASCII字符，而IRI可以支持其他字符集，如Kanji。现在人们使用的术语通常是IRI。

下一节。为本体选择哪个选项将取决于您的特定需求以及您的组织或您工作的组织建立的标准。

最后，您应该知道的另一个与名称相关的概念是名称空间的概念。如果您已经使用过大多数现代编程语言，如Python或Java，那么您已经熟悉了名称空间的概念。这个概念在OWL中是相同的。名称空间用于避免不同本体之间的命名冲突。例如，在一个关于电信的本体中，您可能有一个称为“网络”的类。在一个关于图论的本体论中，您可能也有一个名为网络的类。这两个概念虽然是相关的，但却有所不同。与编程语言一样，您使用名称空间前缀来确定名称引用的特定名称空间。例如，在本例中，电信本体可能有前缀tc，图论本体可能有gt。因此，当您为电信本体引用网络类时，您将使用tc：网络并为图论类提供帮助的网络。

请注意，您已经对其他名称空间有了一些经验。OWL名称空间前缀是OWL，用于指诸如owl：东西和owl：没有这样的类。资源描述框架模式（RDFS）是一个OWL构建在其基础之上的模型，因此本体使用的一些属性，如rdfs：标签利用这个名称空间。

在活动本体选项卡的底部视图中，有一个被称为本体前缀的选项卡。此选项卡将显示您的本体中的所有当前名称空间映射。每个本体都需要来自OWL、RDF、RDFS、XML和XSD的某些概念，所以这些名称空间默认会被映射到每个新的Protege本体中。对于本体的名称空间，也有一个到空字符串的映射。这允许您显示和引用本体中的实体，而不需要输入名称空间前缀。如果您现在看这个选项卡，您应该会看到一个行，其中第一列是空白的，而第二列有本体的基本IRI。它应该与Active本体选项卡顶部的本体IRI是相同的IRI，除了它在最后也有一个#符号。例如，为本教程开发的披萨教程有一个IRI of <http://www.semanticweb.org/pizzatutorial/ontologies/2020/PizzaTutorial> and在本体前缀中有空白第一列的行具有IRI：
<http://www.semanticweb.org/pizzatutorial/ontologies/2020/PizzaTutorial#>.

第八章：一个关于一些个体的更大的本体论

本教程的其余部分需要将一些数据加载到本体中。到目前为止，我们主要处理的是定义类和属性。这种类型的信息在语义网社区中被称为T-Box信息。T代表术语。个人或实例被称为A-Box。A代表在关于域的特定事实中的断言。通常，A-Box的信息会比T-Box多得多。A-Box信息通常从电子表格、关系数据库或其他来源上传。本教程中没有介绍的一个有用的工具是Cellfie。Cellfie是一种工具，它可以从电子表格中获取数据，并将其上传到本体中，并将基于表的数据映射到对象和属性值中。有关Cellfie的教程，请参见：

<https://github.com/protegeproject/cellfie-plugin/wiki/Grocery-Tutorial>

除了使用Cellfie之外，您还可以使用第5章中介绍的按类划分的个人选项卡来创建新实例，并为这些实例创建对象和数据属性值，就像您在第6章中对热和中等个人所做的那样。然而，这可能是乏味的，所以没有你的无趣的工作，我已经开发了一个版本的披萨本体，它已经创建了许多人。这个本体应该与您迄今为止开发的本体相同，除了有许多其他的个体。您可以找到这个已填充的披萨本体at：<https://tinyurl.com/PizzaWDataV2>转到这个URL，并下载文件到您的本地机器，然后使用文件>Open。在此之前，关闭当前文件可能是一个好主意，这样您开发的Pizza本体和带有额外数据的新本体之间就不可能出现混淆。

8.1 熟悉这个更大的本体论

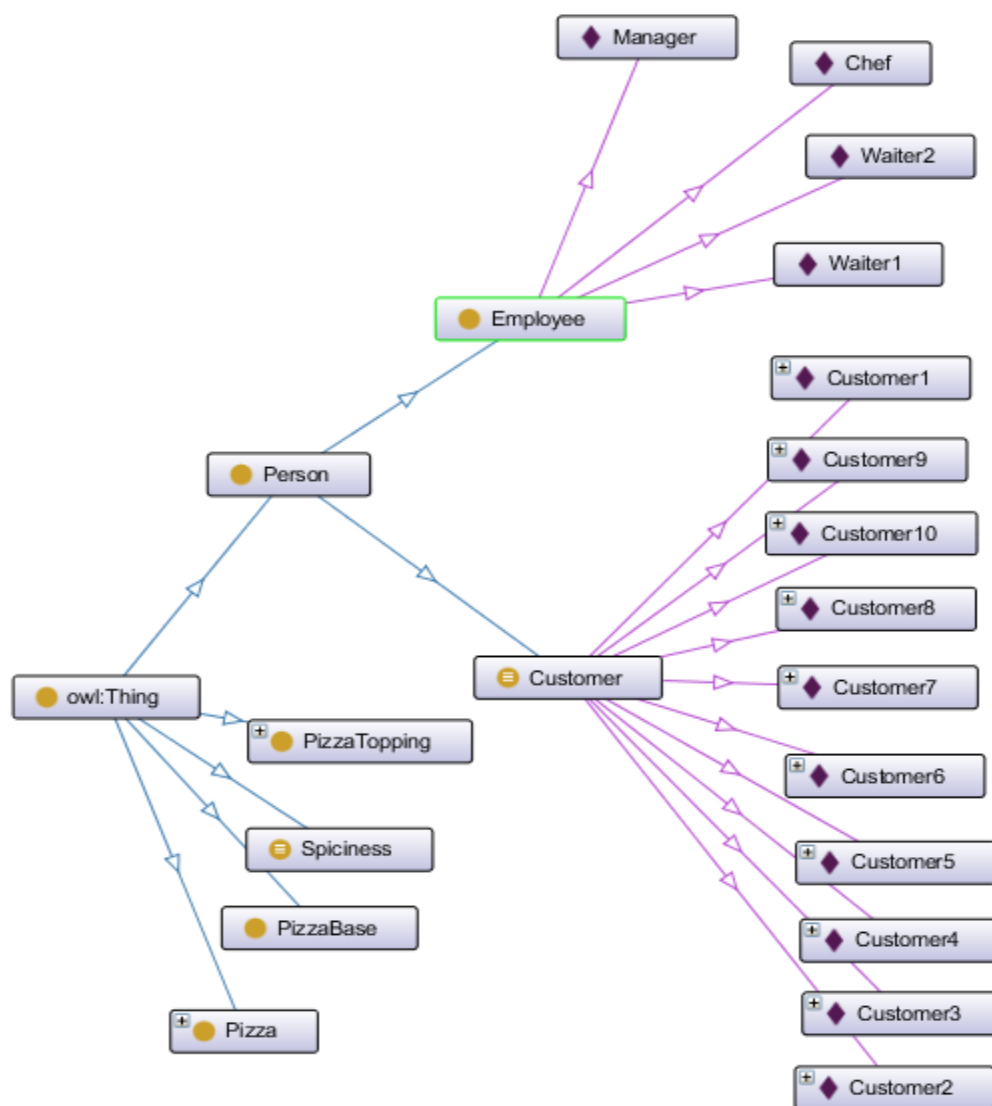


图8.1一些新的本体类和个人的图表

图8.1使用OntoGraf选项卡来可视化本体中的一些新添加的内容。有一个新类称为子类员工和客户。员工有5个个人：经理、厨师、服务员1和服务员2。客户有10个实例。

此外，如果您查看对象属性选项卡，您将会看到有一些新的属性：

物业购买的客户有域名披萨和范围客户。它从一个单独的披萨映射到购买它的顾客。它有一个逆词，叫做购买披萨。

属性有香料，偏好有域，客户和范围的香料。它记录了顾客对他们通常喜欢的披萨的辣度的偏好。

“数据属性”选项卡还将显示了一些新属性：

has折扣数据属性具有一个客户的域和一个范围的xsd：十进制。这将记录客户在下次购买时将获得的折扣（如果有的话）。

购买的数据属性有一个客户域和一个xsd：整数。它记录了每个客户购买的披萨的数量。

ssn属性有一个“雇员”域和一个范围为xsd：字符串。它从一个员工映射到他们的社会安全号码。在美国，这是所有雇主必须处理保险费和税务信息的数字。

hasPhone数据属性有一个人的域和一个xsd：字符串的范围。

大多数这些数据属性除了其范围外，都有其他的约束。例如，折扣只能在0到1之间，并且一个电话号码和社会安全号码必须对应于某种格式。

许多这些约束都可以通过定义范围的DL公理来表示。但是，由于下面将讨论的原因，通常最好使用SHACL语言来表示数据完整性约束，而不是作为DL公理。一般的经验法则是，DL公理用于推理，而SHACL用于数据完整性约束。当然，这就引出了这样一个问题：推理和完整性约束之间的区别是什么？这种区别本质上是模糊的。然而，我们将在关于SHACL的章节中讨论一些指导方针，我们希望这将有助于阐明其中的区别。

最后，“按类查看“个人”选项卡将有助于理解本体中的附加数据。如果你去看那个标签，你会看到很多新的人。除了员工和顾客之外，还有披萨类的例子。您可以在右上角的“个体按类型（推断）”视图中看到所有这些个体。

现在有了更多的实例，您可以看到个人按类型（推断）视图的值。您可以展开和收缩各种类，并查看它们的实例¹⁰。请注意，第4条

高卡披萨也是披萨的实例，但它们没有显示在披萨下，因为所有的高卡路披萨的实例总是披萨类的实例。只显示了披萨类的一个实例，因为所有披萨的其他实例也是披萨的子类的实例，所以它们显示在这些子类下，而不是在披萨下。如果有两个或两个以上的类是一个实例，它们不是彼此的子类，那么它们都将被显示出来。例如，玛格丽塔披萨1是玛格丽塔披萨和低热量披萨的一个实例，它出现在每个类下，因为它们都不是另一个类的子类。披萨有可能是低热量比萨，而不是玛格丽塔披萨，反之亦然。

¹⁰ 请注意，如果您在个人按类型（推断）视图中选择了一个类的实例，则您将无法折叠该类。例如，在图8.2中，我们将无法崩溃员工类，因为它选择了其中一个实例（Chef）。要折叠它，只需选择一个不同的实例，它不是您要折叠的类的实例。

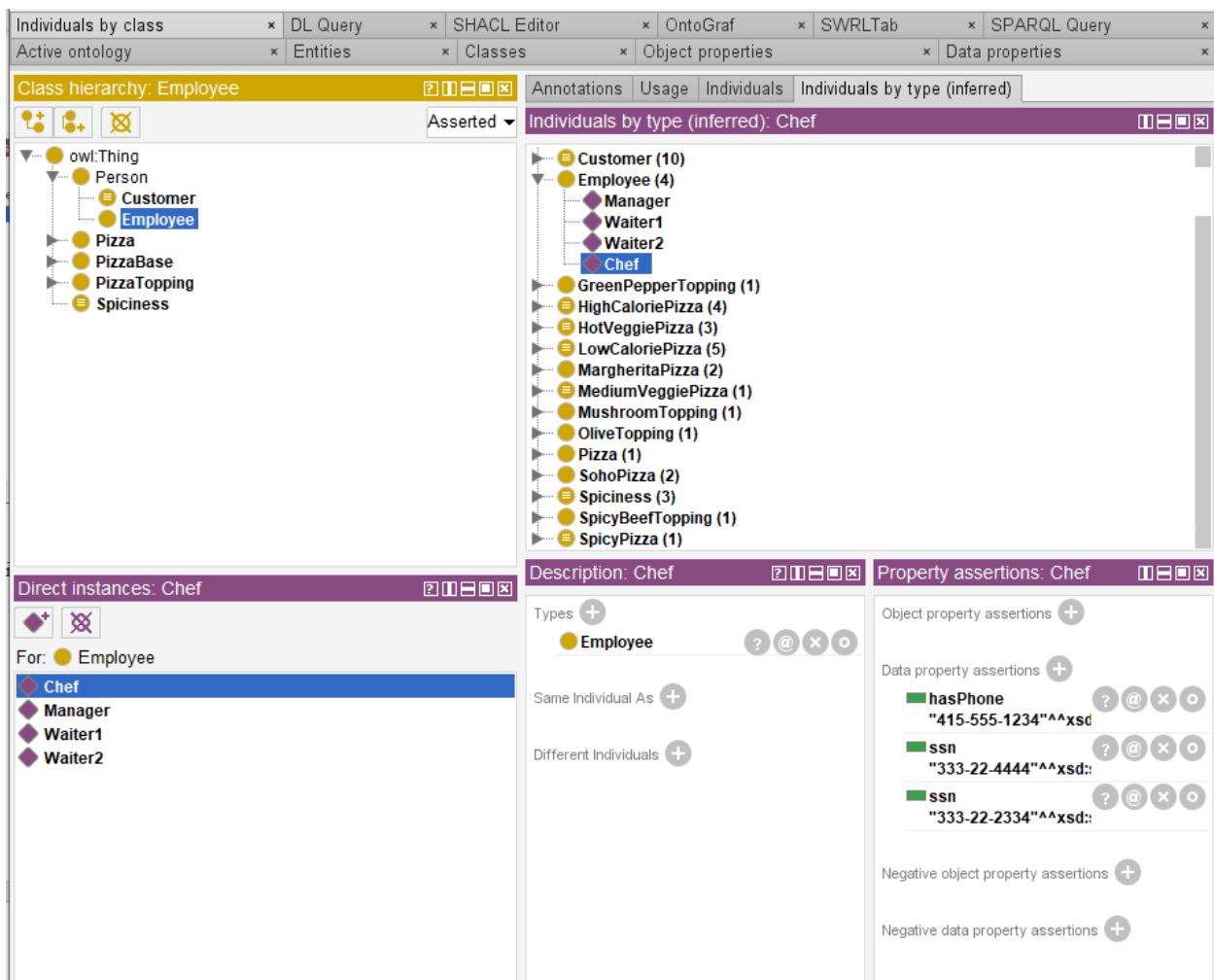


图8.2在“个人按类”选项卡中查看新实例

第九章查询：描述逻辑和SPARQL

现在我们的本体中有了一些个体，我们可以做一些有趣的查询。在Protege中有几个工具用于执行查询。

9.1 描述逻辑查询

首先，根据您已经学到的内容，最直接的查询是描述逻辑（DL）查询。这些本质上是您用于定义类的相同语句。但是，除了使用这样的语句来定义一个类之外，您还可以将其用作查询。

练习33：尝试进行一些描述逻辑查询

-
1. 首先，请导航到“DL查询”选项卡。如果它不存在，创建它使用：**窗口>选项卡>DL查询。**
 2. 在这个选项卡的右上方，您应该看到一个表示DL**查询：下面的视图Query（类表达式）。**
 3. 您可以在此框中输入您想要的任何DL语句，然后查看作为它的子类、超类和实例的所有实体。例如，**输入：顾客并购买一些披萨（超过一些（香料值热））。**也就是说，所有购买了热辣披萨的顾客。起初，你可能什么也看不见，但不用担心，还有一步。
 4. 请查看“查询”下右边的复选框。**检查超类、子类（尽管默认情况下应该已经检查过）和实例。**现在你的UI应该如图9.1所示。你可能会注意到那个猫头鹰：没有什么会显示为一个子类。别担心，这是预期的。记住那个猫头鹰：没有什么空集合，空集合是每个集合的子集（包括它本身），所以就像猫头鹰一样：东西是每个类猫头鹰的超类：没有什么每个类的子类。如果你不想看到猫头鹰：没有什么你可以取消勾选框的显示**猫头鹰：没有。**
 5. 尝试一些额外的DL**查询，例如：一些（值热）和素食披萨和一些（值热）。**请注意，在这最后一个查询中，您正在利用您为第6章中的香料类的实例定义的传递顺序。
 6. 您还可以查询实体名称中的字符串。例如，首先在查询窗口中对Pizza**进行查询。**然后在“名称包含”**字段中输入热。**这应该给你所有的类和个人与热在他们的名字。
-

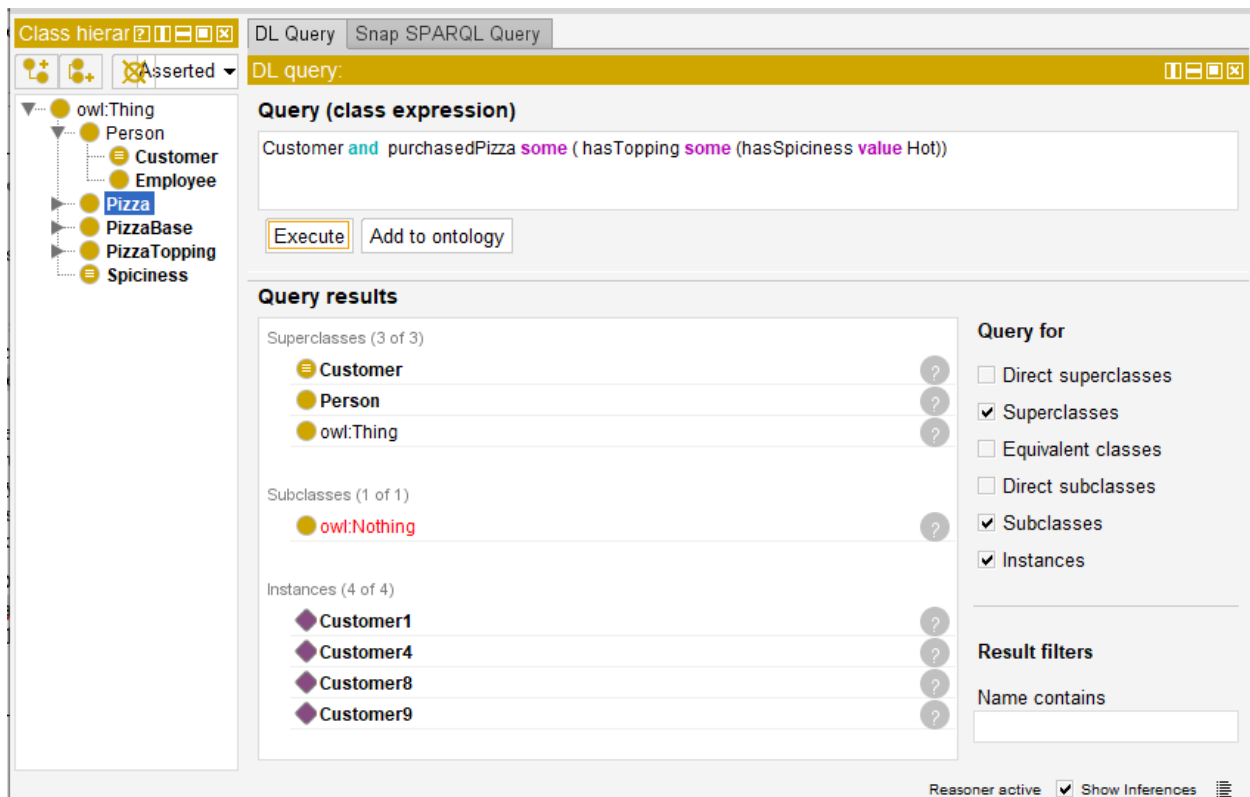


图9. 1关于DL查询的选项卡

9. 2 SPARQL查询

SPARQL是一种强大的语言，人们可以写一本书。事实上，也有关于它的书写过。我见过的最好的一本是鲍勃·杜查姆的《学习SPARQL》。这是一本优秀的书，不仅介绍了SPARQL，还涉及了RDF/RDFS以及如何使用三元组来表示OWL中的所有信息。我将在这里只谈这些问题，还有很多关于它们要说的，杜查姆的书是一个学习更多的好地方。如果以下内容中的一些内容有点难以理解，请不要气馁。这只是试图对一些需要重要的研究才能真正理解的东西进行非常高水平的介绍。

SPARQL本质上是语义Web和知识图，就像SQL是关系数据库一样。正如SQL可以做的不仅仅是查询一样，它还可以在数据库中断言新的信息，所以SPARQL也可以。当前SPARQL插件蛋白质有些有限，不支持语句如插入输入新数据所以我们将覆盖的基本使用SPARQL作为查询语言，但记住有很多比我们简要介绍。

9. 21 一些SPARQL披萨查询

首先，请进入“**SPARQL查询**”选项卡。如果它还不在那里，你可以像往常一样添加它使用窗口>选项卡>SPARQL查询。这个选项卡由两个视图组成，顶部保存查询，底部保存结果。应该已经有一些文本了。这可能看起来很令人困惑，但我们会解释一下。**首先**，点击选项卡底部的执行按钮。您应该看到返回了一堆类和类表达式。

要了解发生了什么，您首先需要了解每个SPARQL查询都由两部分组成。开头的第一部分由几个名称空间前缀组成。这些语句包括用于特定名称空间的前缀以及与此名称空间相关联的IRI。回想一下，在第7章中描述了这些概念。您可能想知道所有这些前缀都是从哪里来的，因为您没有将它们添加到您的本体中。答案是，每个OWL本体都带有一组定义本体所需的名称空间和前缀。

此外，要理解SPARQL，你需要在OWL中“达到顶峰”。到目前为止，我们一直在讨论纯逻辑术语和集合理论术语中的概念，即在语义层面上。但是，与任何语言或数据库一样，也有一个较低的级别来描述如何将概念映射到实际数据。在关系数据库中，表示数据的基本构造是一个表。在OWL中，基本构造是一个三重结构。OWL实际上是建立在RDFS之上的，而RDFS是一种建立在RDF之上的语言。RDF（资源描述框架）是一种描述图形的语言（在这个术语的数学意义上）。即，用来描述节点和链接。

RDF图的基础是由主语、谓词和宾语组成的三元组。这就产生了所谓的无向图或网络图，因为对象可以是主题，反之亦然。每当您在OWL中定义一个属性时，您都是在定义一个谓词。个人可以是主题，也可以是对象（或两者都有）。例如，在我们的本体中，客户1购买了美国披萨1。在这个例子中，客户1是主题，购买的Pizza是谓词，美国热的Pizza1是对象。

但是，类和属性本身也被表示为三元组。例如，当你创建一个“披萨”类时，Protege为你做的是将它添加到三个：Pizza rdf:type owl:类到本体中。也就是说，披萨实体的类型为（是猫头鹰的一个实例）：类。类似地，当你将命名披萨作为披萨的子类添加时，Protege会添加了三重：命名披萨 rdfs: 披萨的子类。

希望现在您可以对这个初始查询进行一些理解。该查询正在查找作为三元组主题的所有实体，其中谓词是rdfs:子类的，并且对象是任何其他实体。这个在名称之前表示名称是通配符，可以匹配符合模式的其他部分的任何内容。这是SPARQL的力量的一部分，一个人可以匹配一个主语，一个宾语，一个谓词，甚至所有三个。将模式通配符的所有三个通配符将返回图中的每三个部分（在这种情况下，我们的整个Pizza本体）。您可能会注意到，在某些情况下，对象只是一个类的名称，而在其他情况下，它是一个类表达式，前面有一个橙色圆圈的前面。这是因为在使用DL公理定义类时，Protege会创建了与各种DL公理对应的匿名类。

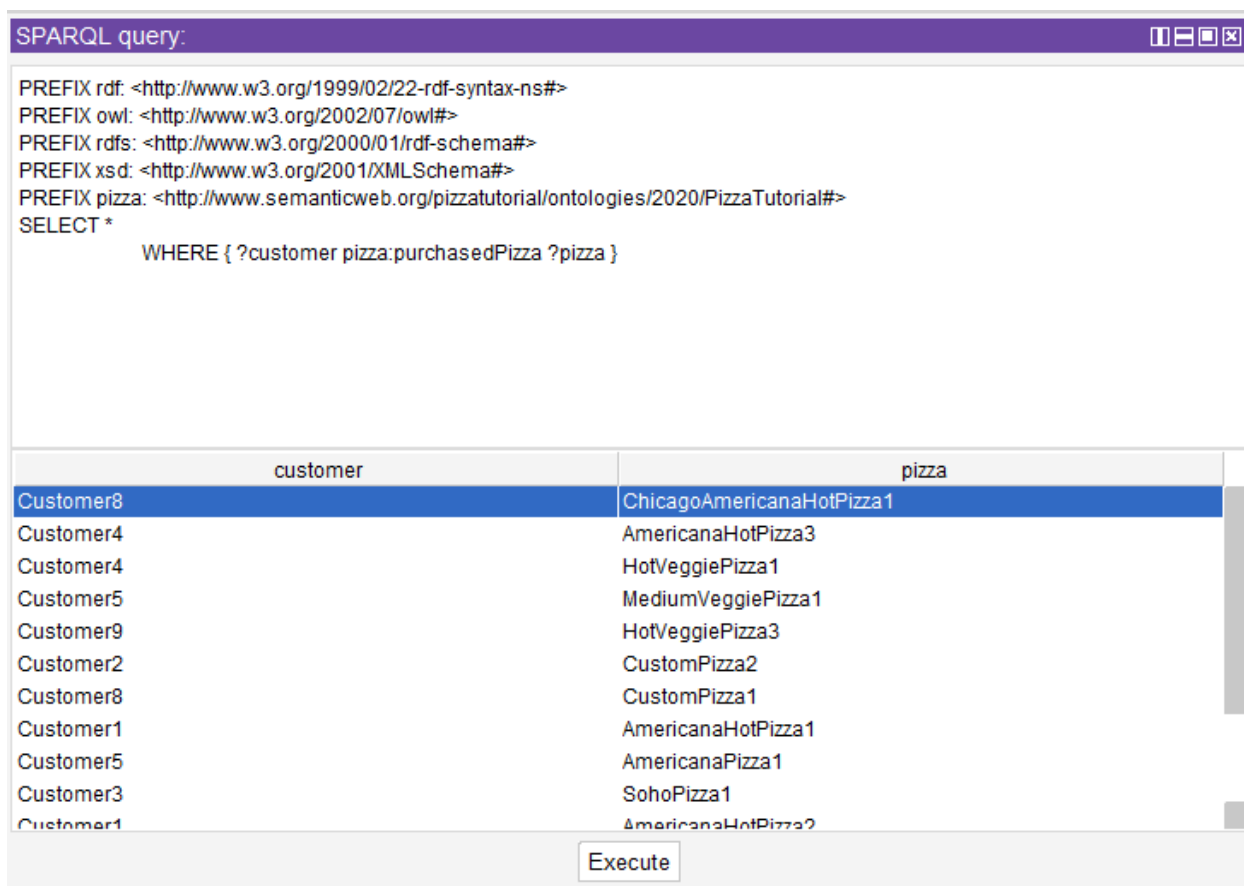
SPARQL查询的选择部分决定了要显示的数据。查询的位置部分决定了在查询中匹配的内容。如果您想在WHERE子句中显示所有匹配的内容，您可以只对选择子句使用*。此选项卡中的初始默认查询是在不了解特定本体的情况下设置的。也就是说，它将返回所有作为其他类的子类的类，而不考虑本体论。要获取关于Pizzas的信息，我们需要做的第一件事是在查询的开头添加另一个前缀。在我们的例子中，Pizza本体已经被映射到前缀披萨（您可以在第7章中讨论的Active本体选项卡的主体前缀选项卡中看到这个）。因此，在最后一个前缀语句后，向SPARQL查询中添加以下内容：

前缀披萨：<<http://www.semanticweb.org/pizzatutorial/ontologies/2020/PizzaTutorial#>>

我们几乎已经准备好查询实际的本体了。对于我们的第一个查询，让我们找到一个客户购买的所有披萨。下面的SPARQL代码是：

选择*地点{ ? 顾客披萨 : 购买披萨 ? 披萨 }

在前缀下面的查询窗口中键入它（当然可以删除现有的查询）。点击**执行**。您的屏幕应该与图9.2相似。



The screenshot shows a SPARQL query window with a purple header. The query is as follows:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX pizza: <http://www.semanticweb.org/pizzatutorial/ontologies/2020/PizzaTutorial#>
SELECT *
WHERE { ?customer pizza:purchasedPizza ?pizza }
```

Below the query, there is a table with two columns: 'customer' and 'pizza'. The results are as follows:

customer	pizza
Customer8	ChicagoAmericanaHotPizza1
Customer4	AmericanaHotPizza3
Customer4	HotVeggiePizza1
Customer5	MediumVeggiePizza1
Customer9	HotVeggiePizza3
Customer2	CustomPizza2
Customer8	CustomPizza1
Customer1	AmericanaHotPizza1
Customer5	AmericanaPizza1
Customer3	SohoPizza1
Customer1	AmericanaHotPizza2

At the bottom of the window, there is an 'Execute' button.

图9.2 A SPARQL查询

如果您仔细检查输出，您可能会注意到一个问题。客户4似乎只买了2个披萨。但是，如果你检查个人**按类选项卡**中的数据，你会看到她购买了3个。其中一个没有显示出来的原因是，当输入数据时，我通常会在客户实例中输入它。然而，对于一个顾客4的披萨，我在披萨上输入了数据。也就是说，我在热素食Pizza2上声称它是由客户客户4购买的。由于购买的披萨和购买的客户是逆，推理机为我填写了额外的信息。但是，SPARQL并不关注推理者所主张的信息，而只关注用户所主张的信息。注意：这取决于SPARQL的实现。例如，还有另一个SPARQL实现可以作为Protege插件称为Snap SPARQL使用，它知道推理机的推断。

但是，在我们正在使用的默认SPARQL选项卡中，SPARQL忽略了推理器断言的信息。这是一个问题，为其他插件的蛋白质学家，如个人矩阵。幸运的是，对于这个问题有一个简单的解决方法，可以保存推理器断言的信息并重新加载，使其与用户定义的数据相同。我的博客在一篇文章中描述了这个解决方法：

<https://www.michaeldebellis.com/post/export-inferred-axioms>

若要进一步查看此内容，请将当前查询（确保保留所有前缀）替换为：

选择披萨：由顾客购买

这将显示您的两个披萨，购买关系是在披萨的实例，而不是在客户的实例。

假设您想看到客户的所有对象吗？有了一些新的构造，这很简单。首先，在SPARQL中，标识任何实体的类型的快捷方式是使用关键字a作为谓词。这只是rdf： type的缩写。其次，当您在WHERE子句中有多个语句时，您需要用一个句点结束每个语句。

替换为当前查询：

选择*

在哪里{？顾客吃披萨：顾客。

客户关系与客户相关。

这将提供图中作为客户类的某个实例的所有对象的长列表。即，作为以客户为主语的谓语对象的任何实体。

假设你想计算一下到目前为止顾客购买的披萨的数量。为此，您可以使用SPARQL函数计数。下面是它的样子：

选择（计数（？披萨）AS？pcount）

顾客披萨：购买披萨

将其粘贴到SPARQL查询视图中，**然后**点击执行，您应该会看到返回的值： 15。然而，请记住，这并不是所有的披萨，因为一些购买被记录在披萨上，而不是顾客上。为了得到完整的号码，我们可以利用我们已经记录了每个客户购买的披萨的数量，并使用了SPARQL的SUM功能。该查询将是：

选择（SUM（PNUM）ASPUSM）

顾客披萨：购买的披萨号码

这应该给你正确的17。

9.22 SPARQL和IRI 名称

如果您还记得，在本教程的开头，我们将创建新实体的首选项更改为用户提供的名称，而不是默认自动生成的名称。现在您已经看到了使用SPARQL的例子，我们可以解释原因了。使用自动生成的名称，而不是有名称，如客户或购买，我们的实体将有名称，如OWLPropertyA4257yri73ff90rmbx和OWLClass23gkb0tk5kd30tm。因此，第一个查询将是类似于以下内容的查询：

选择*{？客户披萨：OWLPropertyA4257yri73ff90rmbx？piasa}，第二个查询是：

选择*

在哪里？顾客披萨：拥有23gkb0tk5kd30tm。

客户关系与客户相关。

这将远不如用户定义的名称那么直观。使用自动生成的名称是有很好的，特别是对于用多种自然语言实现的大型本体。然而，对于新用户，特别是那些计划使用SPARQL和SHACL的用户，我认为如果需求显示用户提供的真正需求，从用户提供的名称开始，然后自动生成名称。这种逐步开发软件的方法，而不是试图设计出能够扩展到未来所有可能需求的完美系统，这被称为软件开发的敏捷方法。根据我的经验，敏捷方法已经在无数的现实项目中证明了自己，可以比其他瀑布式方法按时、预算交付更好的软件。更多关于敏捷方法see：<https://www.agilealliance.org/agile101/>

这只是给您一个SPARQL可以完成的一些事情的基本概述。还有很多，如果你感兴趣，你应该看看杜Charme的书或一些SPARQL工具和教程。其中一些是在参考书目中找到的。

最后一点：OWL和SWRL的新用户经常感到沮丧的特性是开放世界假设（OWA）和缺乏非单调推理。OWA在第4.13章中讨论了。非单调推理将在第11.1节中进行讨论。现在，尽管请记住，SPARQL不受这些限制中的任何一个。使用SPARQL，人们可以进行非单调推理，并利用更常见的封闭世界假设（CWA）。例如，您可以测试特定实例上的属性的值是否存在，如果该属性不存在，则可以采取操作。

要执行非单调推理，只需要在同一查询中组合插入和删除。一个是删除现有的三元组，并插入具有相同主语和谓词但具有不同对象的三元组。不幸的是，目前不可能在Protege中演示，因为Protege中的实现都不支持插入或删除。我计划在将来的某个时候向我的博客中添加一些SPARQL实用程序，以提供一些例子。关于如何使用SPARQL来绕过OWA的一个例子，请参见我最近添加到我的博客中的本体和说明：

https://www.michaeldebellis.com/post/the-people_example-ontology在这个例子，我展示了定义的类隐士如何定义为一个人的社会关系最多为0的人由于OWA不能被推理者识别。然而，一个简单的SPARQL查询既可以识别没有社会关系的People类的实例，也可以通过添加三重元素来改变此类实例的类型（?p）：

?p rdf:type Hermit。这篇新文章也有一个更有趣的例子，SPARQL查询与Franz Inc. 及其寓言图开发的产品。该查询利用链接数据，通过查询几个像DBpedia和Geonames这样的不同的公共知识图，来查找巴拉克·奥巴马出生的地区的收入中位数。

第十章SWRL和SOWL

创建语义Web规则语言（SWRL）是因为有某些类型的推断是不能通过描述逻辑（DL）公理来完成的。此外，根据我的经验，也可以使用DL完成推理，但将推理定义为规则可以更直观。

在Protege中，实际上有两个针对SWRL的UI。有SWRL选项卡，还有一个规则视图，可以在我们在8.2节中添加一个视图时添加到UI的规则视图中。SWRL选项卡正在进行更积极的开发，我建议您始终使用它。本章将重点介绍SWRL选项卡。本章中的所有内容都适用于SWRL选项卡，并且在Rules视图中会略有不同。有关规则”视图的概述，请参见本章末尾列出的SWRL流程建模教程。

与所有规则系统一样，SWRL由左侧（称为前行）和右侧（称为后续）组成。这两个字符之间由一个用破折号和一个大于像这样的字符创建的箭头分隔：->。SWRL规则中的每个表达式都用一个^符号分隔。当且仅当前词中的每个表达式都得到满足时，规则的结果才会触发。由于前因可以多次满足，这意味着SWRL规则可以进行迭代。它们将触发所有能够满足前项的值的组合。所有参数（通配符且在规则触发时动态绑定的变量）前面都有？。

SWRL表达式包括3种类型：

1. 类表达式。这是一个类的名称，后面跟着一个带有参数的圆括号。例如，“客户”（?c）将把?c绑定到“客户”类的实例，并且（假设前项的其余部分已满足）将遍历“客户”类的每个实例。
2. 属性表达式。这是一个属性的名称，后面跟着括号和两个参数：第一个用于被测试的个体，第二个用于绑定到该个体的该属性的值。请注意，由于个体可以为一个属性拥有多个值，这也可以创建迭代，其中规则将遍历每个个体的每个属性值。例如，购买披萨（?c、?p）将绑定到每个客户购买的每个披萨。
3. 内置函数。SWRL有许多内置的函数，用于进行数学测试、字符串测试等。SWRL内置是记录的here：<https://www.w3.org/Submission/SWRL/>所有SWRL内置是由swrlb前缀。例如，内置的数学如果?np的值大于1，则大于（（?np，1）成功。

我们将在披萨本体中添加两个简单的SWRL规则，来计算一些客户的折扣。我们假设我们的披萨店营业时间不长，所以他们想给任何购买了多个披萨的人一个折扣。此外，他们的经理高估了顾客对辛辣食材的喜爱，他们有很多墨西哥胡椒，所以他们为喜欢热披萨的顾客比喜欢中号或温和的顾客提供更大的折扣。

首先，让我们制定第一条规则，给所有购买了2个以上披萨并更喜欢热披萨的顾客打20%的折扣。

练习34：编写你的第一个SWRL规则

1. 首先导航到或创建SWRLTab。如果它还不存在，请使用窗口>tabs>SWRLTab来创建和选择它。如果你没有在窗口>选项卡菜单下，然后使用文件>检查插件，并选择SWRLTab插件。记住，如果你这样做，你需要重新启动Protege，以使插件可用。

2. SWRLTab分为两个主要视图，然后是选项卡底部上与机器人相关的一些按钮。何时以及如何使用机器人工具的问题让许多新用户感到困惑，但有一个简单的答案：不要使用它！随着你对SWRL有更多的经验，你将会开始理解如何以及何时使用DR00LS，但是对于初学者来说，答案很简单。我认为所有这些滑稽的按钮只供高级用户使用。你根本不需要使用它们。这就是为什么我们在第4.2节中安装了颗粒推理器。Pellet推理器支持SWRL，当您运行推理机时，它也会自动运行任何SWRL规则。关于机器人的一篇论文请参见参考书目。

3. 单击俯视图底部的“新建”按钮。其他的按钮应该是灰色的，因为它们只在您至少编写了一个规则时才适用。这将给您提供一个新的弹出窗口来编写您的规则。在顶部的名称字段中，调用规则：热计算规则。你可以跳过这个评论，但如果你想添加一个评论，这是一个好习惯，你可以写一些这样的东西：为那些喜欢热披萨的顾客提供一个特别的折扣。

4. 现在，请转到规则窗口的底部，并开始编写该规则。要开始，您希望将一个参数绑定到客户类¹²的每个实例。要做到这一点，你所需要做的就是写下：客户(?c)。注意，自动完成应该在这个窗口中工作，但有时可能不会，您可能需要键入完整名称。此外，当您在状态字段输入时，您将在状态字段中看到各种提示或错误消息，目前您基本上可以忽略这些提示。例如，当您输入“客户”时，您会看到诸如：无效的SWRL原子谓词“Cus”等消息，直到您完成客户类的名称。这些消息可以帮助您理解为什么当您开发更多的规则时，您的规则不会进行解析，但现在您应该能够忽略它们。

5. 现在，您希望将一个参数绑定到迄今为止每个客户订购的披萨数量。为此，您首先添加一个^字符。这代表了逻辑上的和。也就是说，该规则将对满足前面内容中所有表达式的每一组绑定进行触发。要测试披萨的数量，您使用数据属性编号购买。所以在这一点上，你的规则应该看起来像：客户(?c)^号码，购买(?c, ?np)。

6. 现在我们想测试对象属性中的香料偏好。第一个参数也将是?c。也就是说，我们正在遍历客户的每个实例，并将其绑定到?c，然后测试这些属性的值。然而，在这种情况下，我们不是将辣度偏好绑定到一个参数上，而只是想测试ifit，而是等于辣度热的实例。所以我们在表达式中直接引用了这个实例，结果是：^的香料偏好(?c, Hot)。

7. 作为前面的最后一部分，我们想测试客户是否购买了超过1个披萨。我们可以使用SWRL数学内置的swrlb：比。添加^ swrlb：希腊比(?np, 1)这是最后一个

¹¹ 更多关于机器人的信息，请参见论文：M. J. O'Connor (2012)。参考书目中的一对OWL 2 RL推理器。

¹² 这实际上不是必需的。在没有客户(?c)表达式的情况下，您将得到相同的结果，但是这是一个很好的例子，说明了如何使用类的名称用SWRL遍历它们的实例。

前面的一部分，所以我们写了->来表示结果的开始。在这一点上，你的规则应该像：客户(?c)^号码，购买(?c, ?np)^有香料偏好(?c, 热)^swrlb:greaterThan(?np, 1)->

8. 最后，我们写下规则的结果，即箭头后面的那部分，表示每次规则成功时该做什么。我们想给这些客户20%的折扣，所以我们写道：有折扣(?c, 0.2)。左边的表达式是测试规则是否应该触发，而右边的表达式是要添加到本体中的新值的断言。对于那些有逻辑背景的人来说，简单的思考方法是前因是隐式普遍量化的，而结果则是隐式存在量化的。

9. 因此，整个规则应该是这样的：客户(?c)^号码购买(?c, ?^)^香料偏好(?c, 热)^swrlb:伟大比(?np, 1)->折扣(?c, 0.2)。请注意，底部的OK按钮只能在规则具有有效语法时进行选择。它现在应该是可选择的，所以可以选择它。您应该会看到新规则显示在最顶部视图的顶部。

请注意，在SWRL中有一个小错误，有时当前本体的前缀会被添加到所有没有前缀的表达式中。所以在某个时候，你可能会看到你的表情最终是这样的：披萨：顾客(?c)。如果发生这种情况，不用担心，它根本不会影响规则的工作方式。如果在某种时候发生这种情况，您想删除前缀，有一种方法在我的blog：<https://www.michaeldebellis.com/post/removing-ontology-prefixes-from-swrl-rules>中描述

接下来，我们想为其他已经点了多个披萨但不喜欢热披萨的顾客编写第二个SWRL规则。

练习35：编写另一个SWRL规则

1. 请确保您仍然在SWRLTab中。单击热计算规则，并选择克隆

2. 这将打开与您在窗口中创建第一个规则的代码相同的窗口。将此规则的名称从S1更改为最小香料折扣规则。

3. 接下来，编辑针对客户的辛辣度偏好的测试。与其只是测试它是否热，我们想测试这次它是否比热。这是一个使用我们在第6章中定义的顺序关系的例子。更改香料偏好(?对香料的偏好(?c, ?spr)。与其只是测试它是否等于Hot，我们还需要将偏好值绑定到参数?spr。然后在这之后添加常规和角色和新的测试，所以你应该添加：^是^(?spr, Hot)

4. 最后，我们想把这些客户的折扣改为10%而不是20%。所以将结果改为：hascrect(?c, 0.1)。

5. 因此，整个规则应该是这样的：客户(?c)^号码购买(?c, ?^)^偏好(?c, ?^)^比(?^, 热)^>:比(?c, 1)->折扣(?c, 0.1)。

现在我们想运行我们的规则。记住，没有必要使用这些机器人按钮。刚才同步推理器和您的规则应该像维护基于值的其他DL公理一样触发

逆、定义类等。回到个人分类标签，查看不同的客户。例如，顾客1已经点了不止一个披萨，而且香料很辣，所以她有0.2的折扣。请注意，与推理者所断言的任何信息一样，有一个↑？在您可以单击的断言旁边，它将提供关于为什么断言该值的解释。此解释将列出触发的适当规则和导致其触发的值。如果你看一下客户6，你会发现他没有折扣，因为他只买了一个披萨。最后，如果你看看顾客2，她的折扣是0.1，因为她买了不止一个披萨，但她的辣度是更辣。

在这种情况下，我们的规则的结果是向个人添加一个数据属性断言。另一个可能的结果是使一个人成为一个新类的实例。例如，如果我们有一个名为优先客户的客户子类，并且我们希望规则的结果是使客户成为优先客户的实例，我们可以有->优先客户（?c）作为规则的结果。

一个对调试松鼠规则很有用的工具是语义查询增强的Web规则语言或SQWRL（发音为SWRL）。SQWRL规则看起来就像SWRL规则，只是在随后有一个sqwrl：select语句，它列出了我们想在每次触发规则时知道其值的每个参数。

练习36：编写一个SQWRL规则

-
1. 打开SQWRLTab if it使用窗口>Tabs>SQWRLTab还不存在。您会看到它看起来几乎与SWRLTab完全相同。
 2. 假设我们想看看热数规则触发的频率。我们很容易找到。要开始选择热计算规则并克隆它。这将创建一个名为S1的规则副本。选择该规则，然后选择“编辑”按钮。
 3. 将规则的名称更改为测试热折扣规则。用下列内容替换后续的（箭头后面的表达式）：sqwrl：选择（?c，?np），然后选择“确定”。你的SQWRL规则应该是这样的：客户（?c）^号码，已购买（?c，?np）^有香料偏好（?c，披萨：热）^swrlb: greaterThan(?np, 1) -> sqwrl: select (?c, ?np) . 同步推理器。
 4. 选择测试热数计算规则，然后选择选项卡底部的运行按钮。这将在下视图中创建一个新的选项卡，称为测试热数计算规则。您应该看到，当?c等于客户4、客户1和客户8以及?np等于3、2和2时，规则触发了3次。
-

这是对SWRL的一个非常简短的介绍。对于一个更有趣的基于流程建模的例子，see：
https://www.michaeldebellis.com/post/swrl_tutorial

第11章SHACL

接下来，我们将看看SHACL的一个插件。SHACL代表形状约束语言。请注意，在此上下文中的形状与几何形状无关。SHACL比这里描述的其他技术要更新一些。然而，它填补了语义Web架构堆栈中的一个重要空白，并且它在大规模企业发展的世界中获得了很大的吸引力。出现SHACL的原因乍一看似乎有点难以理解。毕竟，SHACL可以为数据定义的许多约束也可以使用描述逻辑或SWRL来定义，这些约束更高级，也更容易使用。那么为什么还要使用SHACL呢？SHACL对于在现实世界中使用语义网络 and 知识图技术至关重要，有两个原因：

1. 需要定义不受开放世界假设（OWA）和单调推理限制的约束。
2. 事实上，现实世界的的数据是混乱的！

11.1 OWA和单调推理

如第4.13节所述，OWL使用了开放世界假设（OWA），因为它为互联网而设计的。然而，OWA使某些类型的约束验证变得困难或不可能。例如，通常会有一个数据完整性约束，即所有员工都必须有一个社会安全号。虽然这样的公理可以在OWL中定义，但由于OWA，我们希望它很少工作。OWA意味着在互联网上的某个地方可能有一个社会安全号码，但该系统只是还没有找到它。虽然这是真的，但要验证企业数据的完整性，仅仅说“它就在某个地方”是不行的。为了验证数据的完整性，我们需要能够在不存在需要的数据时发出警告，因此我们需要使用封闭的世界假设（CWA）。

单调推理是OWL和SWRL是基于逻辑的一个副产品。OWL推理者本质上是定理的证明者。如果你曾经研究过数学证明，你就知道证明某物是无效的经典方法之一就是证明某个变量有两个不同的值。例如，if four定理暗示 p 为真和 p 为False，那么我们就有一个矛盾。每个变量只能有一个值。这就是为什么SWRL规则只能向变量添加值，而不是更改它们。欲了解更多信息，请参阅马丁·奥康纳在SWRL上的精彩演讲：

<https://protege.stanford.edu/conference/2009/slides/SWRL2009ProtegeConference.pdf>

这就是为什么SPARQL经常需要被使用，而不是SWRL，尽管SWRL更抽象和更强大。SWRL不支持非单调推理，而SPARQL则支持。类似地，在验证数据时，我们可能希望采取行动来更改它，例如，强制它变成适当的标准格式。要做到这一点，我们需要改变而不是增加一个价值，也就是说，我们需要非单调的推理。

11.2 现实世界是混乱的

使用SHACL的另一个原因是，现实世界中的数据很混乱。在本教程的过程中，您可能经历了一个错误，推理机将您的本体标记为无效。如果你没有，祝贺你，但当你与机器人合作时，你会体验到它。这是使用基于逻辑的语言的另一个副产品。仅仅有一个不一致，就会使整个模型无效。对于一些小的例子来说，这不是一个问题。当一个人有几十个、几百个甚至成千上万的人时，找到并解决问题并不难。然而，当处理大数据时，当你拥有数万、数百万或更多的个人时，不良数据的流行率可能是巨大的。

也就是说，可能在这种情况下，我们永远没有得到数据来满足每个完整性约束，这意味着推理机除了告诉我们本体不一致之外，永远没有任何用处。

因此，SHACL提供了一种定义数据完整性约束的方法，这些约束在一定程度上与OWL和SWRL中可以定义的约束有重叠。例如，两者都可以定义一个特定属性所允许的值的数量。例如，每个员工实例必须只有一个社会安全号（ssn）。如果这被定义为一个DL公理，那么该公理将永远不会针对那些因为OWA而没有ssn的员工。另一方面，如果一个员工意外地有2个ssn值，那么整个本体将会不一致，直到有一个值被删除。另一方面，SHACL可以处理这两个示例，而不是使整个本体不一致，而是记录不同严重级别的警告。

11.3 基本SHACL概念

为了理解SHACL，请记住，OWL底层的语言是RDF，它将图形描述为形式的三元组：主题谓词对象。SHACL也可以在RDF级别上工作，因为一些开发人员可能出于效率的原因，只想使用这个较低的级别。因此，RDF可以验证RDF图和OWL本体。从根本上说，SHACL由两个组件组成：

1. 一个用于定义RDF图上的数据约束的RDF词汇表（其中包括OWL，因为OWL本体是一个RDF图）。
2. 用于将1中定义的约束应用于指定的数据图的推理器，如Pizza本体。

1中最重要的类之一是SHACL形状。SHACL Shape类的一个实例由一组目标和约束条件组成。目标定义了数据约束应用于的RDF图中的哪些节点。对于OWL本体，这通常是一个类的名称，它表明约束适用于该类的所有实例。约束定义约束的特定属性以及实际约束，如最小或最大值数和数据类型。在下面的示例中，目标是披萨本体中的员工类。一个约束示例是，ssn属性必须恰好有一个值。另一个约束示例是，ssn值的格式必须是一个字符串：“NNN-NN-NNNN”，其中每个N必须是一个整数。欲了解更多关于SHACL的信息，请参见参考书目中的参考文献。

11.4 Protege SHACL插件

要开始进入窗口>选项卡，看看你是否有SHACL编辑器作为一个选项。如果你不去，然后去文件>检查插件和选择SHACL4蛋白质约束验证器。您需要重新启动Protege才能看到新的插件，这样就可以保存您的工作，然后退出并启动Protege，并使用数据加载披萨本体。

因为在本教程中编辑SHACL有点复杂，我们只查看一些已经编写的SHACL约束，看看验证器如何处理它们，而不是编写额外的约束。首先将PizzaShapes.txt文件下载到本地硬盘上。这个文件可以在：<https://tinyurl.com/pizzatshapes>上找到，一旦你下载了这个文件，就可以打开SHACL编辑器：

窗口>选项卡>SHACL编辑器。

当编辑器打开时，您将在它中看到一个示例形状文件，但这不是您正在寻找的形状文件。从编辑器中单击选项卡顶部的打开按钮，导航到披萨形状。下载的txt文件。

此文件中有两个形状，一个用于员工类，另一个用于客户类。因此，我们只展开类层次视图中的“人”类。我们将从员工类开始，所以选择该类，这将导致所有员工的实例都显示在它下面的视图中。

对于SHACL编辑器，我们希望中间的底部视图能够占用尽可能多的屏幕空间。因此，我们可以通过单击每个选项卡顶部的X来删除选项卡最右边的两个视图。

然后将SHACL Editor视图拖到左边，这样您就可以看到员工和客户类及其实例。您的UI应该看起来类似于图11.1。

要在SHACL编辑器视图中开始检查该代码。请注意，在开头有一个名称空间列表，类似于SPARQL编辑器中的名称空间前缀。在前缀之后是第一个实际的形状，即雇员的形状。此形状限制了员工实例上的属性值。目标类标识此形状所在的类。下面是各种节点（如图中的节点，SHACL也被表示为三元组），这些节点约束了应用于员工类的各种属性。第一个节点将ssn属性的基数精确地约束为1（minCount1和maxCount1）。下一个也适用于ssn，并进一步约束数据，而不是仅仅说它必须是一个字符串。它必须是一个与模式匹配的字符串：“`^\\d{3}-\\d{2}-\\d{4}$`”这是一个正则表达式，意味着模式必须是3位数字（0-9中的数字字符），后面是破折号，后面是2位数字，后面是破折号，后面是4位数字。

接下来的2个节点将处理hasPhone的数据属性。这个属性必须至少有一个值（尽管可能更多），并且还必须符合类似的模式，即3位后破折号后3位，破折号后4位。当然，实际的电话号码可能更复杂和多样，但这只是一个简单的例子。

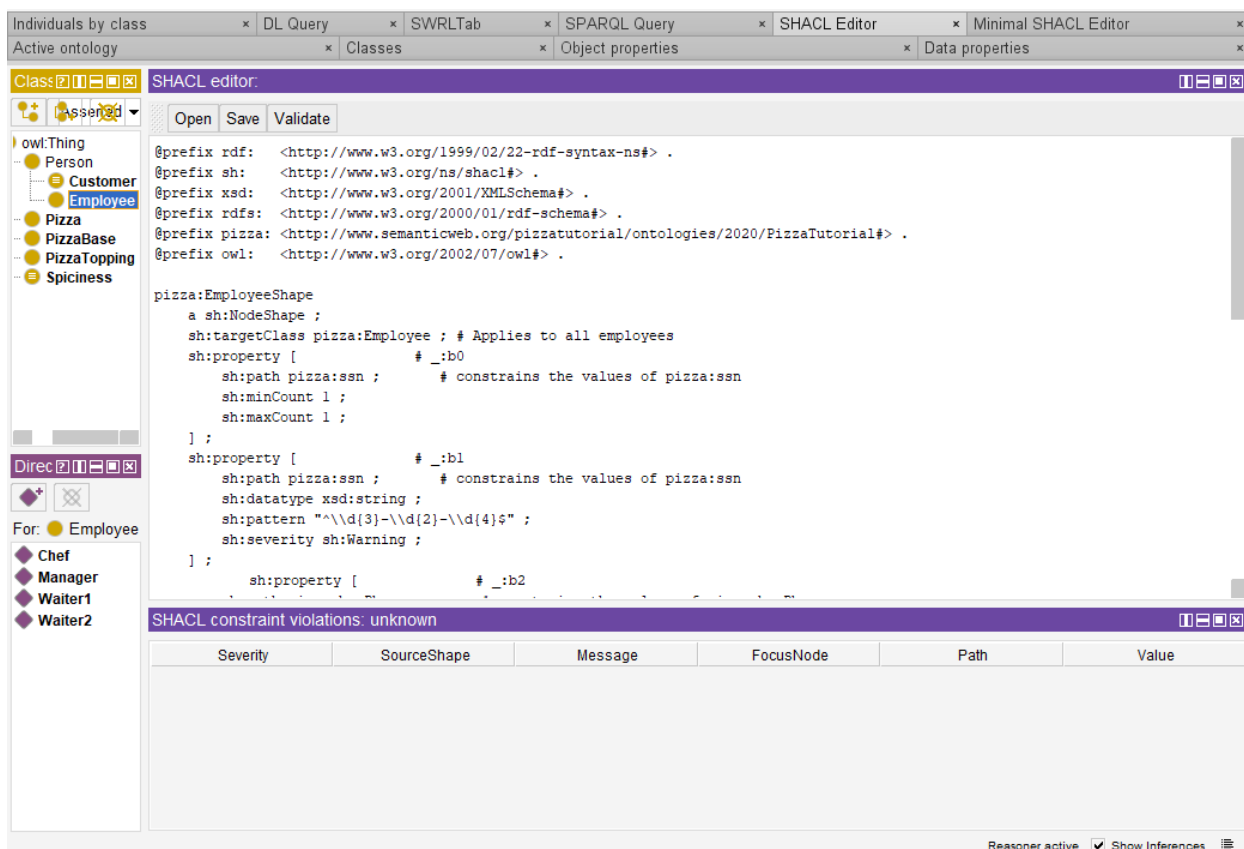


图11.1 SHACL编辑器

现在，点击“验证”按钮。您应该会在底部的长SHACL约束违反视图中看到几条消息。如果您在单击验证时选择了“员工”类，则此视图现在应该是：SHACL约束违反：5/8。这意味着有8个约束违反，其中5个是在员工类的实例上。通过单击每个员工，您可以看到适用于每个员工的违规行为。您可以调整此视图中的各种列的大小，这有助于查看您所需要的信息。最有用的数据是在消息、路径和值列中。所有其他列，如严重性和源性，都可以尽可能小，以便为其他列腾出更多的空间。如果你这样做，并点击主厨个人，你会看到她有一个约束违反。见图11.2。

您可以看到，厨师个人的ssn属性有2个值，这超过了允许的值。如果你在班级视图中检查厨师个人，你会发现情况确实如此。

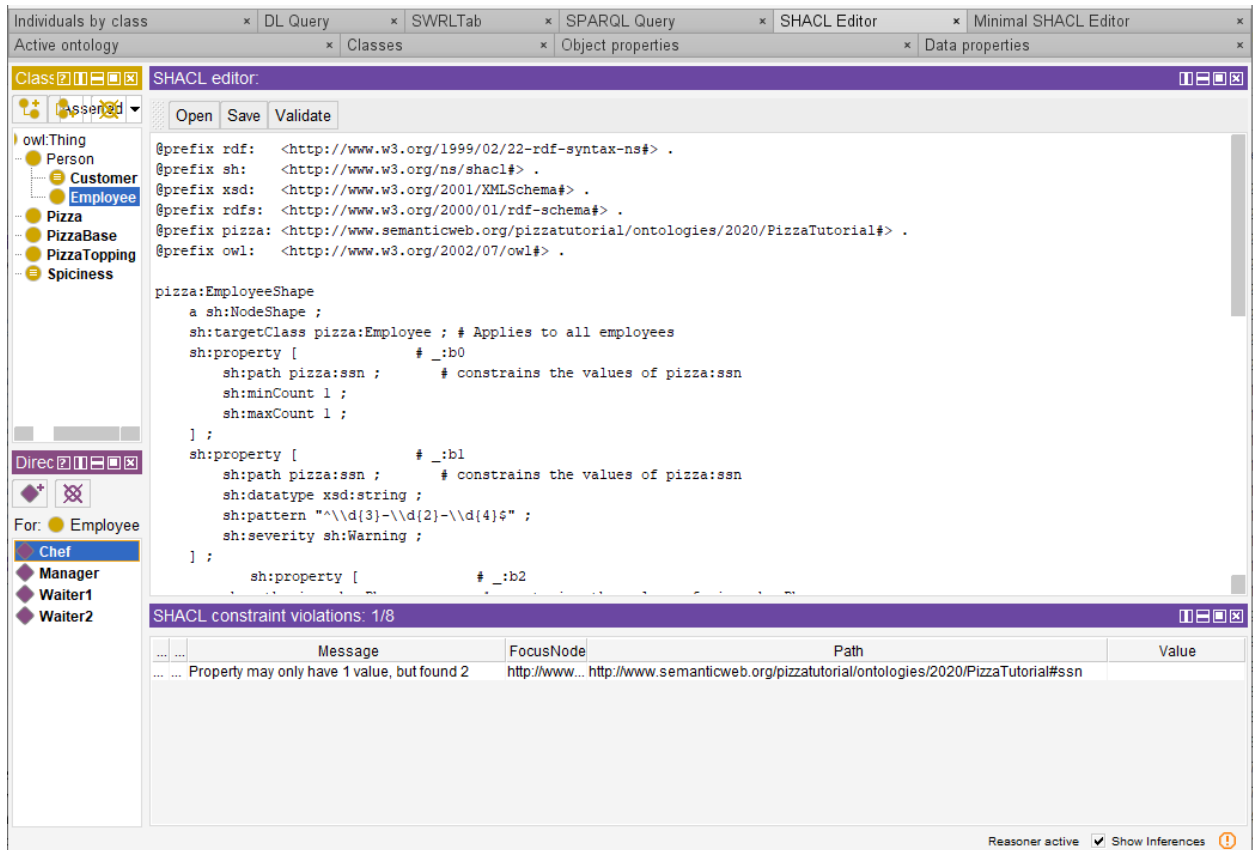


图11.2 厨师个人违反约束的规定

如果您单击经理个人，您将看到他违反了约束条件，因为他的电话号码没有使用正确的格式。服务员1也有类似的问题。等待器2缺少数据。她的手机和ssn数据属性都必须有值，但没有。

如果您将焦点转移到客户类上，则可以看到其余3个约束违规。客户10的折扣属性大于1，这是不允许的。这是由has折扣节点中的客户形状定义的，其中包含sh:最小包含0.0和sh:最大包含1.0。这是为数值属性定义最小值和最大值的方式（注意：这适用于值而适用于值的数目）。客户2也有一个与定义的格式不匹配的hasPhone值，最后，当至少需要一个值时，客户3也没有一个针对hasPhone的值。

回想一下，SHACL约束本身本质上是RDF图。图11.3和11.4说明了上述例子中使用的员工形状和客户形状，如词汇图中的Gruff工具中所述。

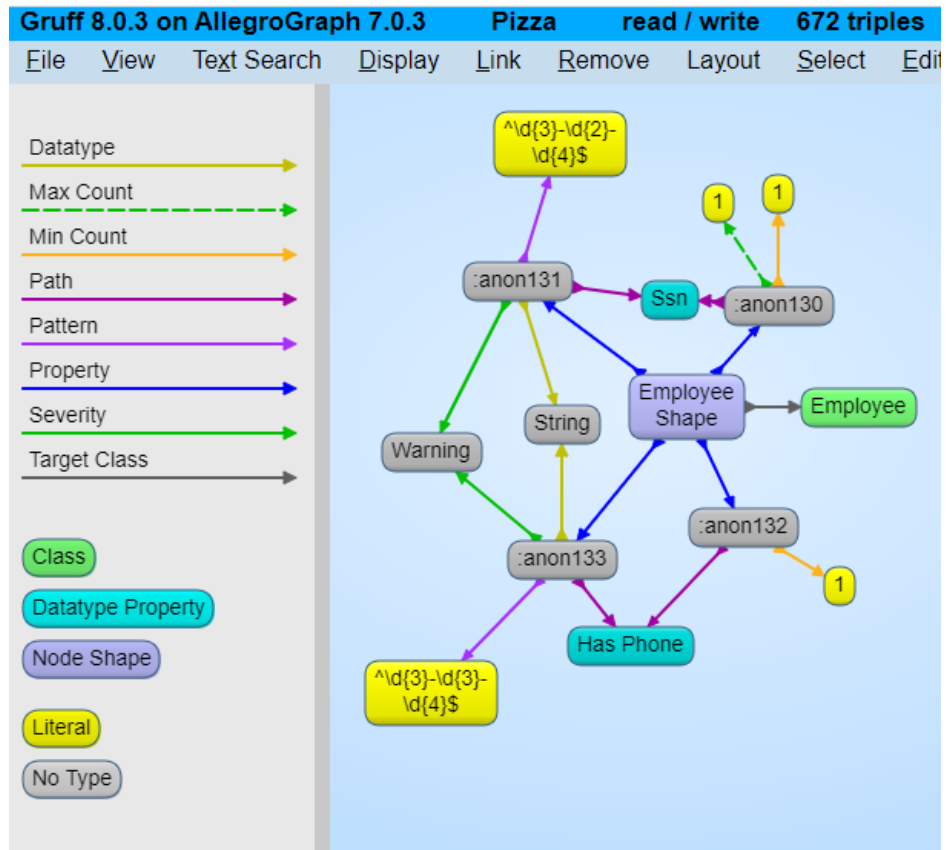


图11.3员工形状的粗糙可视化

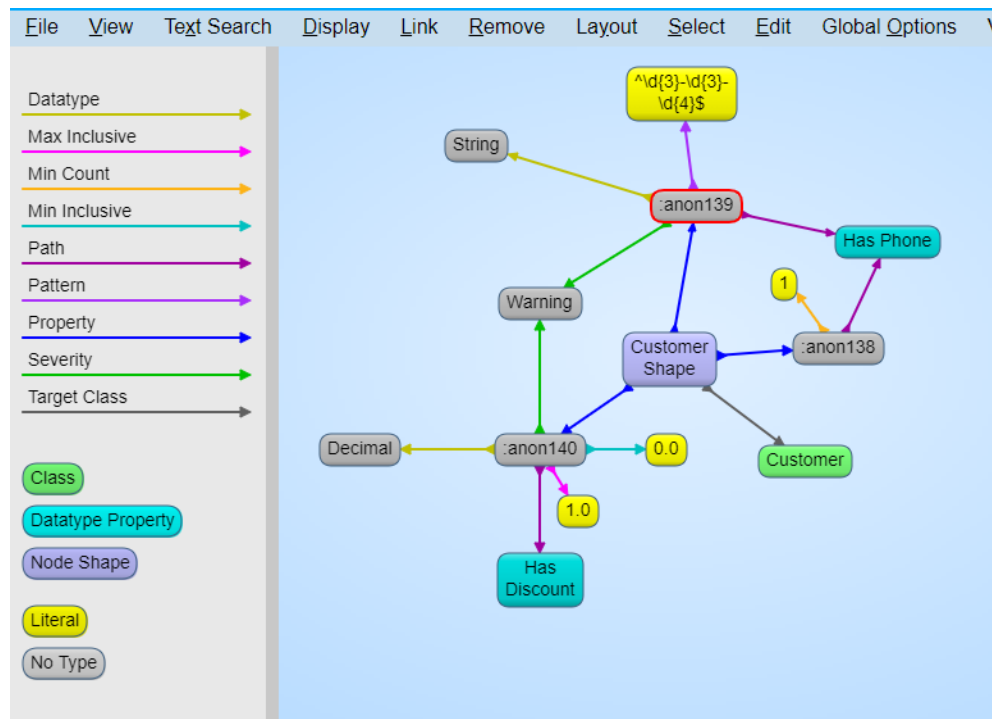


图11.4客户形状的粗糙可视化

这只是对SHACL最基本的介绍。有关更复杂的教程，请参阅顶部象限教程：<https://www.topquadrant.com/technology/shacl/tutorial/Also>，本演示文稿：<https://www.slideshare.net/jelabra/shacl-by-examples>有更多关于SHACL的细节。

第十二章网络抗议

本教程主要关注桌面版本的Protege，因为在撰写本文时，Web程序机不支持任何推理机，所以OWL和Protege的大多数复杂功能，如已定义的类和SWRL规则，都不能在Web程序机中创建。然而，我创建这个教程的目标之一是解决我在Protege用户支持电子邮件列表中经常看到的问题，其中最常见的问题之一是Protege之间的区别。我们都习惯于使用工具作为服务，而不是安装在本地机器上的应用程序，所以人们经常使用Web Protege作为默认设置。虽然Web Protege缺乏推理机支持，但它仍然对协作开发非常有用。本章解释了Web Protege的好处，以及同时使用这两种工具的一些最佳实践。

首先，建议新用户从Protege 13开始。由于缺乏推理器而对本体施加的约束是很重要的，如果你只学习使用Web Protege，他们将会错过OWL的许多好处，并最终认为Protege只不过是一个传统的对象建模工具。然而，一旦人们熟悉了桌面版本，就值得熟悉Web Protege，因为它对于2个或更多的人联合开发本体非常有用。

首先，网络产品有两个选择：

1. 使用斯坦福服务器atwebprotege.stanford.edu
2. 在您自己的本地服务器上下载并安装Web Protege软件。

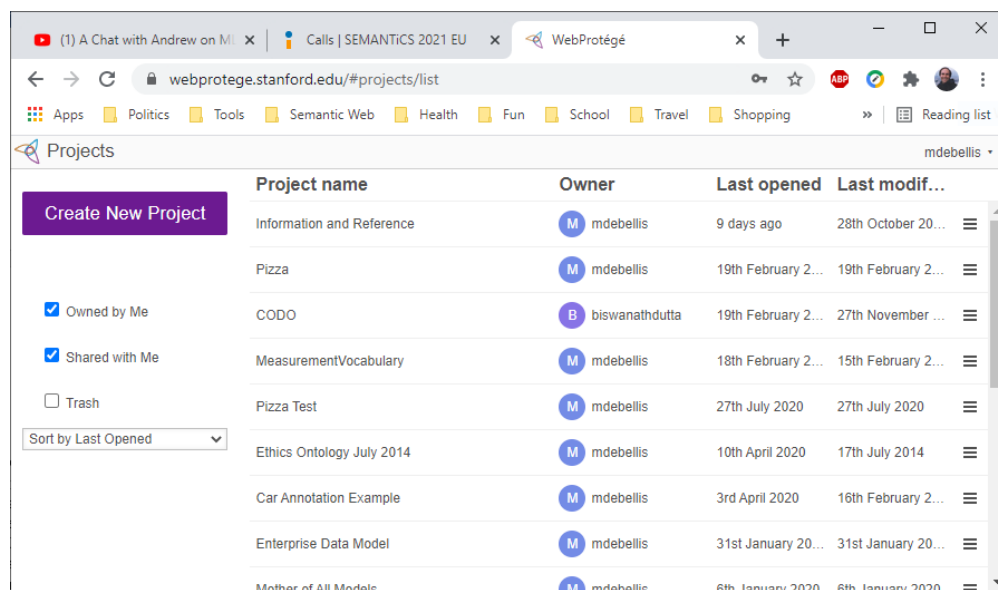


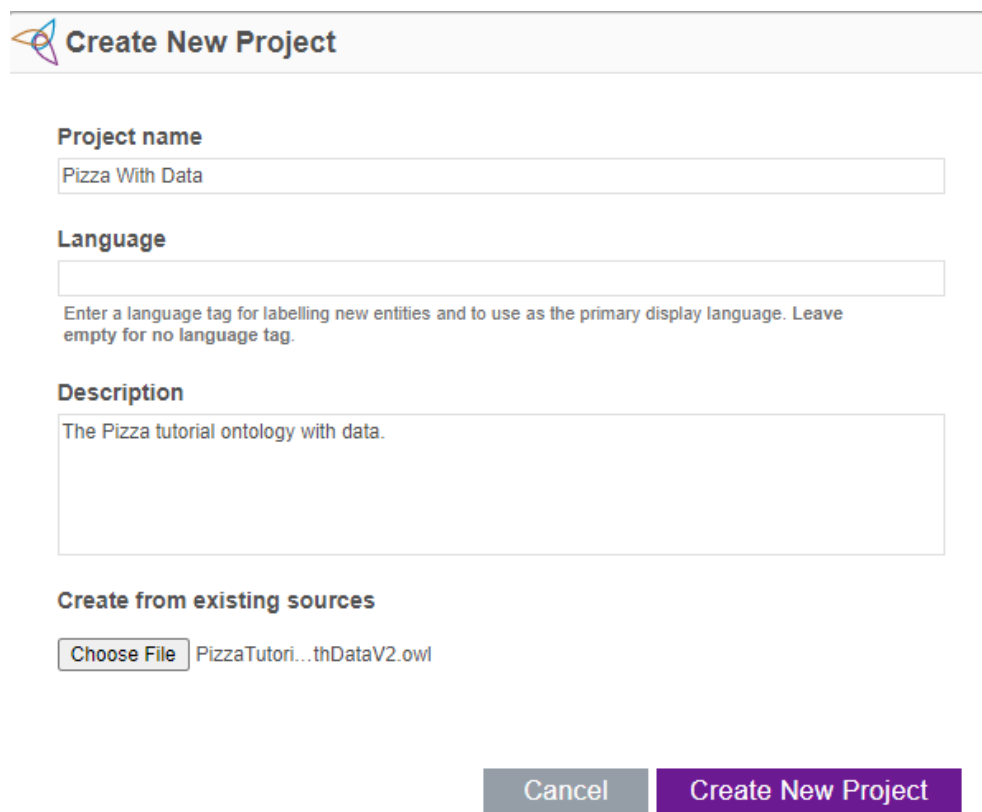
图12.1 Web探测器项目

除非您的组织中有安全要求，禁止您使用斯坦福的托管版本，否则最好从斯坦福服务器开始。我不会详细介绍如何安装本地Web Protege服务器，因为这两种方式的功能都是相同的。首先，让我们在Web Protege中创建一个带有数据本体的披萨教程的项目。当你第一次去[twebprotege.stanford.edu](http://webprotege.stanford.edu)时

¹³ 从这一点上开始，我将把桌面版本的Protege称为只是Protege。

系统将提示您创建用户ID（电子邮件地址）和密码。一旦你这样做了，你应该有一个新的Web程序工作空间。图12.1显示了我的Web程序仪工作空间目前的样子。大多数项目都是由我所有的，不过请注意，COD0项目是由我的同事比斯瓦纳斯·杜塔所有的。然而，由于比斯瓦纳特配置我的访问权限为能够查看和编辑本体的方式，我仍然能够完全访问该本体。

要上传Pizza本体，请选择较大的“新建项目”按钮。这将打开图12.2中所示的窗口。填写项目名称和描述，然后选择“选择文件”按钮，并导航到您有最新版本的披萨教程的数据。注意，在图中，我已经完成了这个导航，所以有一个文件的加载值。您可以将“语言”字段留空。设置了类似于图12.2的所有字段后，单击此对话框上的“新建项目”按钮（注意，这与开始时的按钮不同）。



Create New Project

Project name
Pizza With Data

Language
Enter a language tag for labelling new entities and to use as the primary display language. Leave empty for no language tag.

Description
The Pizza tutorial ontology with data.

Create from existing sources
Choose File PizzaTutori...thDataV2.owl

Cancel Create New Project

图12.2 “新建项目”对话框

您的工作区现在应该包括您的第一个项目。单击项目最右边的三个水平条。这应该会弹出一个弹出式菜单。选择“打开”选项。这将使您进入主Web Protege UI来浏览本体。

在对本体进行更改之前，您需要确保新实体和渲染的设置与用于Pizza本体的设置一致。Web Protege和Protege一样的默认设置是使用自动生成的uuid，而不是用户提供的名称。如果你不确定这些设置，你可以回到第4章和第7章开始的练习2来刷新你的记忆。使用自动生成的uuid有很好的理由，但是对于初学者来说，特别是对于那些想要学习SPARQL的人，我认为他们使学习基础知识变得更加困难，所以我们一直在使用用户提供的名称的替代方法。在Web程序UI的右角有

各种链接：显示，项目，共享，.....单击项目。这将给你一个下拉菜单。从该菜单中选择“设置”。向下滚动到“新的实体设置”。将IRI后缀从自动生成的通用唯一ID（UUID）更改为“提供的名称”。保持其余设置，并选择屏幕右下角的“应用”按钮。

选择“应用”时，应返回主“蛋白质”视图，并选择“类层次”选项卡。如果它没有选择该选项卡。选择披萨班。你的UI应该如图12.3所示。

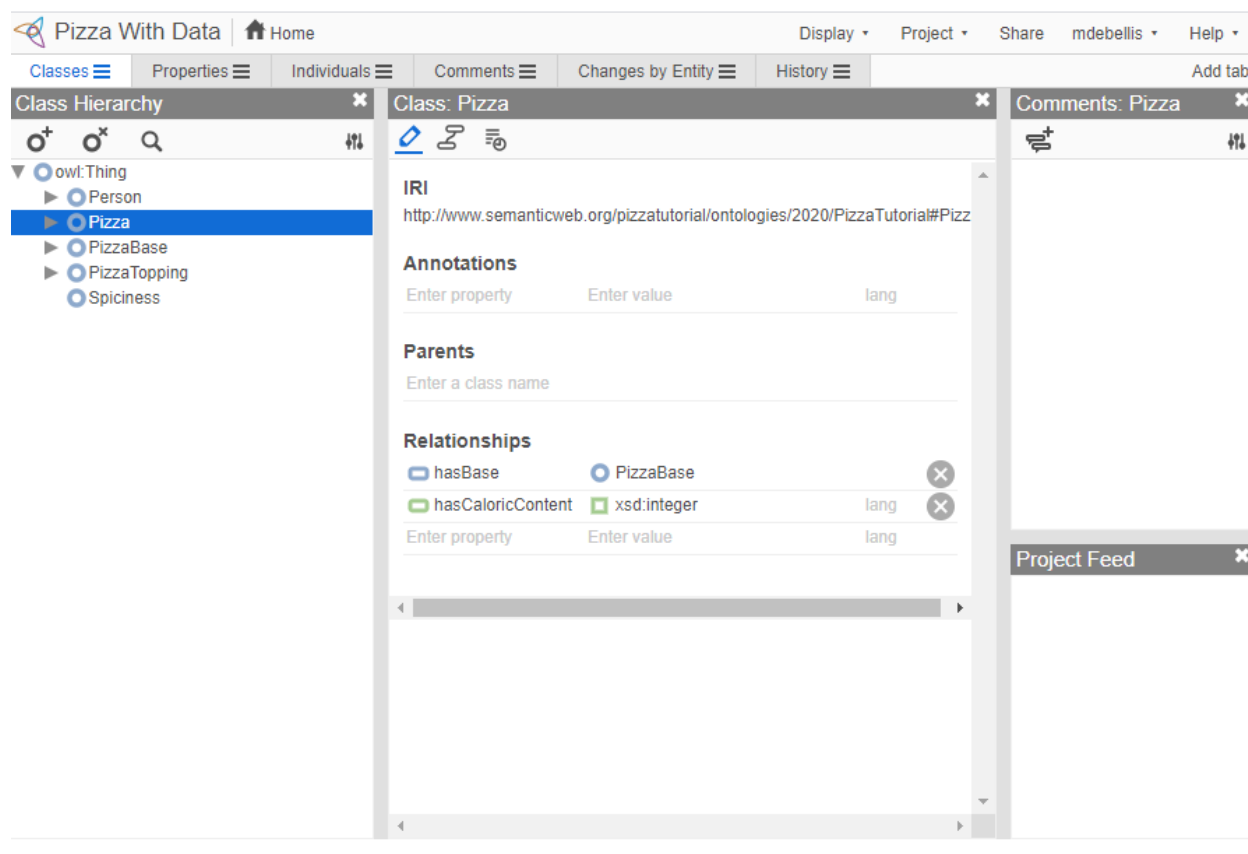


图12.3浏览网络程序仪中的披萨本体

请注意，披萨类的公理存在的，但它们是不可编辑的。Protege使用推理器即使输入或编辑公理来确保任何公理有适当的语法因为没有推理机在Web探测器是不可能编辑公理（因为一个可能引入语法不正确的公理）只删除他们使用X的每个公理。

还要注意右边的面板：评论和项目提要。这些是Web程序中的新功能，以促进本体的协作设计。我们将很快演示这一点。首先，您可以通过单击每个至少有一个子类的类的左侧的三角形来导航类的层次结构。这将扩展该类并显示它的子类。点击披萨来查看它的子类。根据我们的设想，想象一下我们将在芝加哥的披萨店开一家新的分店，我们正在与芝加哥披萨的领域专家打交道。当他们检查等级制度时，他们震惊地发现芝加哥披萨没有阶级，这是一种最初在芝加哥流行的深盘披萨。点击命名披萨。然后单击“注释”视图左角的图标。如果你将鼠标悬停在这个图标上，你就可以做到了

应该看到提示启动新线程，指示此图标的功能。确保名称披萨仍然被选中，然后单击图标以启动一个新的线程。这将打开一个名为“编辑”的窗口，在那里您可以开始一个新的线程。输入类似的内容：我们需要一个命名披萨的子类，叫做芝加哥披萨。它应该有一个公理，要求它有一个DeepPanBase。然后按下“OK”按钮。您的UI应该看起来与图12.4相似。

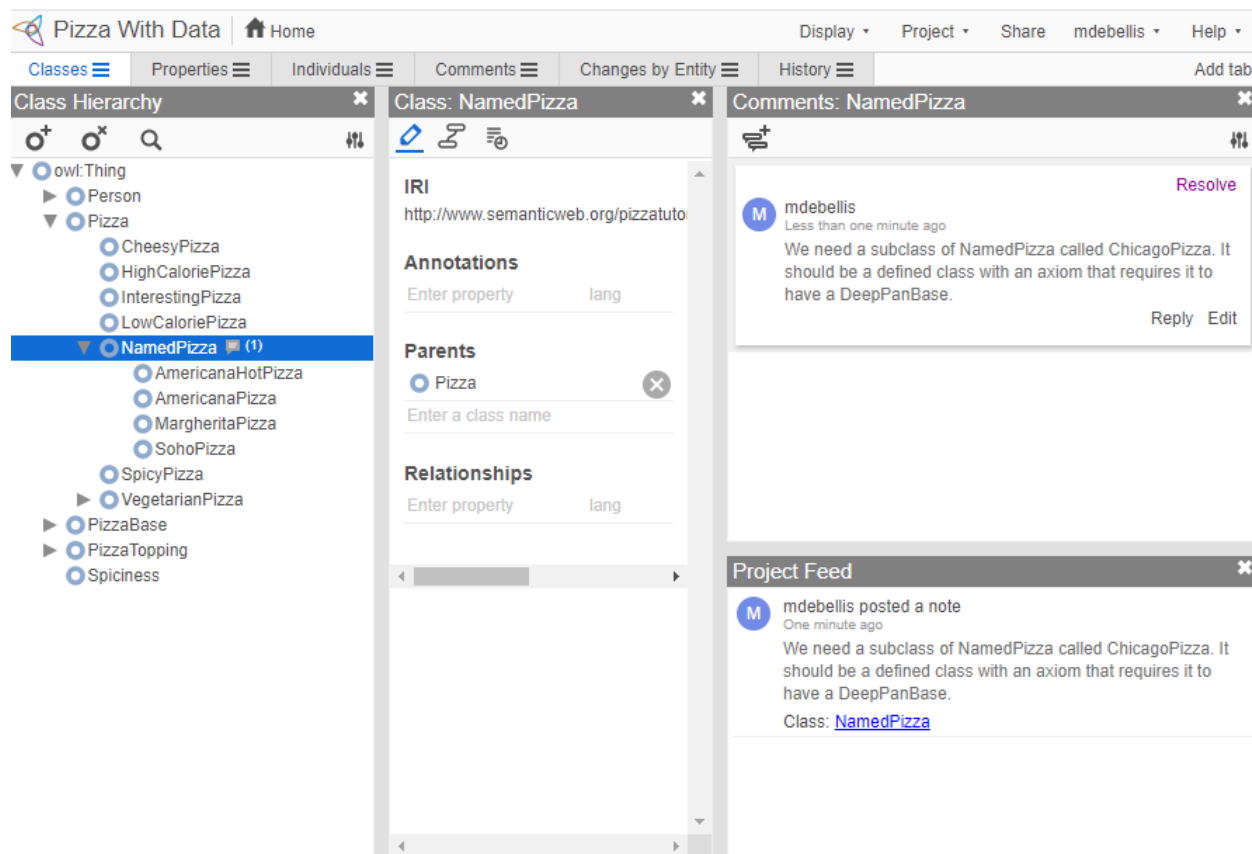


图12.4正在创建一个讨论线程

请注意“注释”视图中的新注释和“项目源”视图中的更新。“注释”视图捕获对本体中的任何实体进行的注释。“项目提要”视图捕获对本体所做的每个更改。

要查看线程讨论是什么样子，请点击新的评论并选择回复。补充一句：我同意，芝加哥披萨很棒，值得上自己的课。

让我们通过创建一个新的命名Pizza子类来解决这个问题。在“类层次结构”视图中选择“名称披萨”。注意这个视图右上角的图标：0+。当您鼠标悬停在此图标上时，它应该说“创建”。您可以使用此图标来创建新的类和子类。确保名称披萨仍然被选中，并单击此图标以创建命名披萨的新子类。这将出现一个名为“创建类”的对话框。在“类名”字段中输入名称“芝加哥披萨”。您可以将“语言标记”字段保留为空。单击“创建”按钮。你现在应该把芝加哥披萨看作是命名披萨的一个新的子类。注意，在项目提要的顶部应该有一个新条目，用于创建这个新类。

您可能会想解决这个线程（注意初始[注释顶部](#)的解决链接），但是，我们还没有真正完成。请记住，我们不仅需要创建这个类，还需要定义一个芝加哥披萨必须有一个DeepPanBase的公理。由于我们不能在Web Protege中添加公理，所以我们需要将本体导出回Protege。通常，我们在导出之前会收集更多的注释和更改，但我们想演示在Protege和网络Protege之间的往返编辑是如何工作的。我们当然可以出口本体从Web产品，然后创建另一个新项目，但这将是麻烦的不断创建新项目每次你想改变产品，如果我们这样做了，我们将失去我们的审计跟踪的评论和变化。幸运的是，有一个更好的方法。

首先，我们需要将本体导出到一个文件中。请注意，顶部的其中一个选项卡是“历史记录”。选择该选项卡。此选项卡显示了本体的每个版本的列表。应该有2个版本标记为R1和R2（在每个版本的右角）。最新的版本总是在顶部，因为这通常是您想要的，尽管也可以将更改回滚到以前的版本。我们想导出最新版本的R2。单击R2图标。这应该会给你一个下拉菜单，其中有两个选项：[还原修订2和下载修订2中的更改](#)。选择[下载版本2](#)。这将提示您使用操作系统的标准文件浏览器，以便使用新的本体保存一个zip文件。本体使用zip文件保存，因为本体可能很大，而且由于Web Protege是在网络上工作的，因此我们可能希望限制大型本体的网络流量。选择适当的位置保存Zip的存档文件。做一些标准的事情来解压缩文件并将其加载到Protege中。注意，当您解压缩该文件时，它也将创建一个目录，因此该文件不会直接位于您保存到的任何目录下。相反，将会有一个名为类似于带有数据本体的披萨-碗-修订版-2的目录，OWL文件将会在其中。

将下载的文件载入到Protege中。转到“[类层次结构](#)”选项卡，并导航到“[命名披萨](#)”下的新芝加哥披萨类。添加公理（如果您需要记住如何在类中添加公理，请参考第4章）。保存文件。现在回到网络程序和你的披萨本体版本。请注意，在窗口的右上角有一些链接（下拉菜单），如显示和项目。选择“项目”，并从下拉菜单中选择“[应用外部编辑](#)”。这将给您一个名为上传本体的小对话框，带有一个小按钮来选择文件。单击“[选择文件](#)”。这将为[您提供一个选择文件的标准操作系统对话框](#)。导航到从Protege保存的文件，然后选择该文件，然后选择“[确定](#)”。这将导致一个名为“[合并本体](#)”的新弹出窗口，在那里您可以看到更改（在此情况下只添加了芝加哥披萨公理），以及一个可以描述更改的文本框。添加适当的提交消息，或仅采用默认值并选择“[确定](#)”。您应该得到一条消息，说明更改已成功应用。

如果你回到芝加哥披萨，你应该会发现它现在有了这个公理。你也可以[导航回命名披萨](#)。在最右边的一栏中，你应该会看到关于需要添加芝加哥披萨作为一个子类的评论。现在已经[完成了](#)，你可以点击评论线右上角的解析链接，评论将从命名Pizza中删除。

第十三章结论：一些个人的想法和意见

本教程只是一个技术的入口点，正在进入高德纳技术炒作周期的启蒙斜坡[高德纳炒作周期]。蒂姆·伯纳斯-李早在2001年就在语义网络上发表了论文[伯纳斯-李2001]。至少根据我的经验，对于大多数美国大公司来说，对机器学习的兴奋似乎暂时掩盖了人们对OWL、SPARQL和其他美国语义网络技术的严重兴趣。然后谷歌[Singhal 2012]创造了“知识图”这个术语。在那之后，脸书[2013年]，亚马逊[海王星2017年]，和许多其他技术领袖开始接受这项技术。企业界终于意识到，机器学习和知识图是互补的，而不是竞争性的技术。在我最近与比斯瓦纳特·杜塔（即将发表）撰写的一篇文章中，关于知识图如何应对新冠肺炎19大流行，绝大多数使用知识图开发的系统利用机器学习完成诸如从“字符串转换为事物”等任务。

术语知识图本身可以用不同的方式使用。我听过的最好的定义是，本体提供了词汇表（即，本质上是T-Box），而知识图是一个与数据结合的本体（A-Box）。尽管在企业界，我经常听到人们简单地谈论知识图，而对词汇表和数据之间的区别不太感兴趣。

有许多新兴的供应商正在以非常高效的方式使用该技术，并为联邦知识图提供基础，这些知识图可以扩展到数亿个三倍或更多，并为所有公司数据提供一个框架。我在参考书目中列出了几个，但这些只是我有一些经验的那些。我相信还有很多其他的人。我使用过的经验最好的产品之一是来自Franz公司的Gruff可视化工具。虽然快板是一个商业工具，但免费版本支持商业版本的大部分核心功能。我发现快板三重节很容易在Windows PC上使用Docker工具来模拟一个Linux服务器。

我第一次开始使用基于分类的语言是我在信息科学研究所（ISI）工作，并使用织机语言[Macgregor 91]为美国国防部及其承包商开发B2B系统。从那时起，我就一直在关注该技术的进步，特别是DARPA的知识共享倡议[Neches 91]，并一直认为这项技术有很大的前景。当我第一次发现探测仪时，这是一次很棒的经历。它是我见过的支持最好、最有用的免费工具之一，让我惊讶的是，没有更多的企业用户在主要方面利用它。我认为我们终于开始看到这种情况发生了，我希望本教程能以一个小的方式帮助我们加速对这个强大而健壮的工具的采用。

第十四章书目

本节不是一个标准的参考书目，而是根据资源分为各种类别，这些资源对未来探索本教程中描述的技术和方法很有价值。

14.1 W3C文件

OWL 2 Primer : <https://www.w3.org/TR/owl2-primer/>

OWL 2技术规格 : <https://www.w3.org/TR/owl2-overview/>

面向对象软件Developers : <https://www.w3.org/TR/sw-oosd-引物/>

SPARQL Specification: <https://www.w3.org/TR/sparql11-query/>

SWRL规范和Built-ins : <https://www.w3.org/Submission/SWRL/>

14.2 网站、工具和演示文稿。

敏捷的Alliance : <https://www.agilealliance.org/agile101/>

Cellfie: <https://github.com/protegeproject/cellfie-plugin/wiki/Grocery-Tutorial>

加纳的炒作周期 : <https://www.gartner.com/en/research/methodologies/gartner-hype-cycle> Jena : 面向语义Web和链接数据应用程序的开源Java框架 : <https://jena.apache.org/>

尼克德拉蒙德和罗布希勒的开放世界假设（OWA）演讲：

<https://drummond/presentations/OWA.pdf> Protege: <https://protege.stanford.edu/>

抗议者的最佳实践。在我的博客上总结了我所有关于Protege，OWL，SWRL等的文章。：
<https://www.michaeldebellis.com/post/best-practices-for-new-protege-users>

SHACL Playground: <https://shacl.org/playground/>

马丁·奥康纳的SWRL演讲：

<https://protege.stanford.edu/conference/2009/slides/SWRL2009ProtegeConference.pdf>

WebProtege: <https://webprotege.stanford.edu/>

WebVOWL：基于web的Ontologies : <http://vowl.visualdataweb.org/webvowl.html> 可视化

14.3 论文

Berners-Lee (2001)。语义网：一种对计算机有意义的新的Web内容形式将释放一场新可能性的革命。还有詹姆斯·亨德勒和奥拉·拉西拉。《科学美国人》，5月17日，2001。<https://tinyurl.com/BernersLeeSemanticWeb>

麦格雷戈，罗伯特（1991）。"使用描述分类器来增强知识表示"。IEEE方面的专家。6（3）：41—46。doi:10.1109/64.87683<https://tinyurl.com/MacGregorLoom>

Neches, 罗伯特 (1991)。支持知识共享的技术。理查德·菲克斯, 蒂姆·芬宁, 托马斯·格鲁伯, 拉梅什·帕蒂尔, 泰德·参议员和威廉·t·斯沃特。人工智能杂志。第12卷, 第3号, (1991)。 <https://tinyurl.com/DARPAKnowledgeSharing>

诺伊, 娜塔莎出版社 (2019年)。行业规模的知识图: 经验教训和挑战。高玉清、安舒耆因、纳拉亚南、帕特森、杰米泰勒。ACM的通信。第62卷。不

8. 八月的2019. <https://tinyurl.com/ACMKnowledgeGraphs>

M. J. O' Connor (2012)。一对OWL 2 RL推理器。与A.K. Das。鹰: 经验和方向 (OWLED), 第九届国际研讨会, 赫拉克利安, 希腊, 2012. http://ceur-ws.org/Vol-849/paper_31.pdf

辛格尔, 阿米特。(2012)。介绍知识图: 事物, 而不是字符串。谷歌高级副总裁, 工程。2012年5月16日。 <https://www.blog.google/products/search/introducing-knowledge-graph-things-not/>

14.4 书

杜查姆, 鲍勃 (2011)。正在学习SPARQL。奥莱利的媒体

刘易斯, 哈利。(1997)。计算理论的要素。和克里斯托·帕帕迪米特里欧在一起。普伦蒂斯大厅; 第二版 (1997年8月7日)。ISBN-13: 978-0132624787

塞加兰, 托比 (2009)。编程语义Web: 使用图形数据构建灵活的应用程序。和科林·埃文斯和杰米·泰勒在一起。奥媒体; 第一版 (2009年7月28日)。

14.5 供应商

三孔图 (Franz Inc.) : <https://franz.com/>

亚马逊的Neptune : <https://aws.amazon.com/neptune/>

码头工人 : <https://www.docker.com/>

Dynaccurate: <https://www.dynaccurate.com/>

Ontotext: <https://www.ontotext.com/>

“池Party : <https://www.poolparty.biz/>”

Stardog: <https://www.stardog.com/>

顶部Quadrant : <https://www.topquadrant.com/>