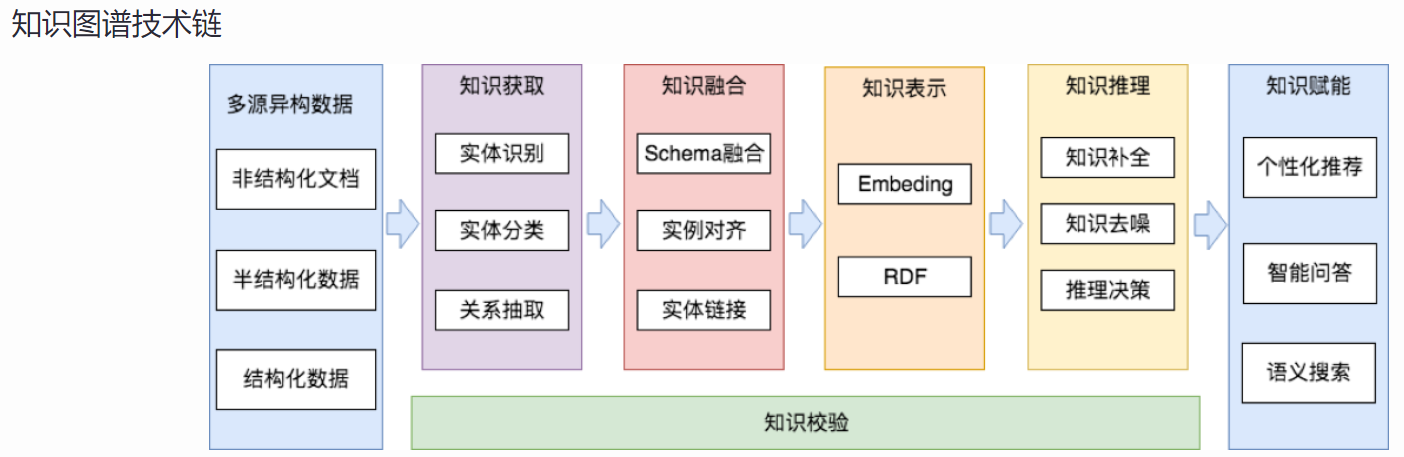
# 知识图谱

知识图谱专注领域：关系分析（个体和个体之间的关系）

本质：多关系图（由多种类型的节点和边组成的图）

如下图所示，知识图谱技术链



利用知识图谱中知识推理环节，来判断、识别错误的关系，可以应用到金融欺诈中去。

关系抽取目的是抽取句子中已标记实体对之间的语义关系。

# 为什么是本体，而不是其他？

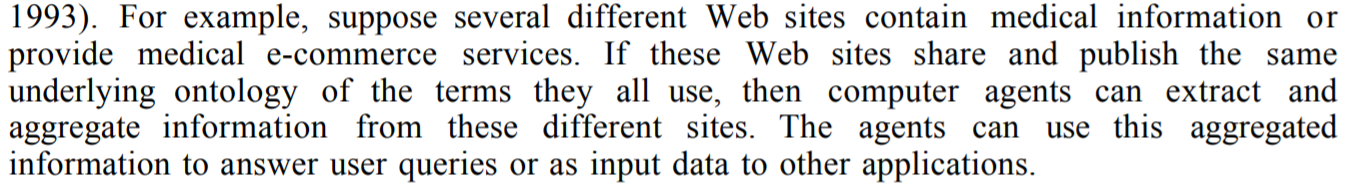
Gruber 1993对本体的定义：领域中对terms明确且正式的描述+terms之间的关系。

网络和现实世界中的存在大量本体，这些本体是有利用价值的。

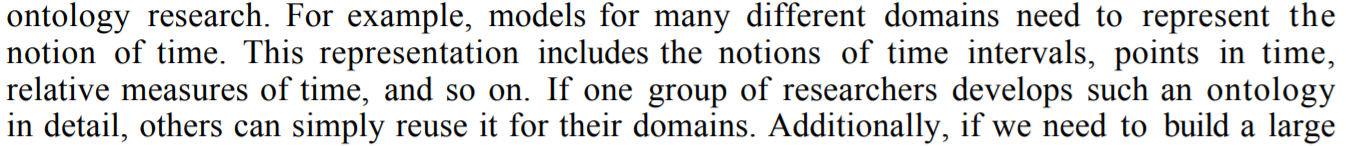
在过去十几年很多机构，致力于标准化本体来更好地分享和标记的信息（并让这些信息能成为机器可识别的）。例如：RDF语言的出现的目的是将知识进行编码，让机器更好得读懂知识。

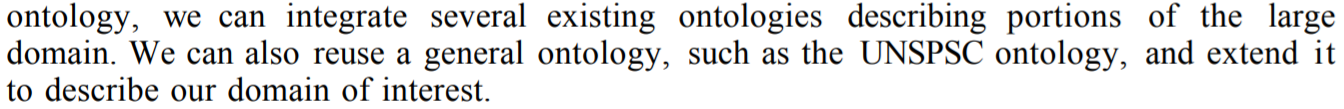
本体存在的意义是

1，将可识别的概念和关系分享给机器或人。概念和关系可以理解为结构化的信息（最重要的意义）。【信息的利用】

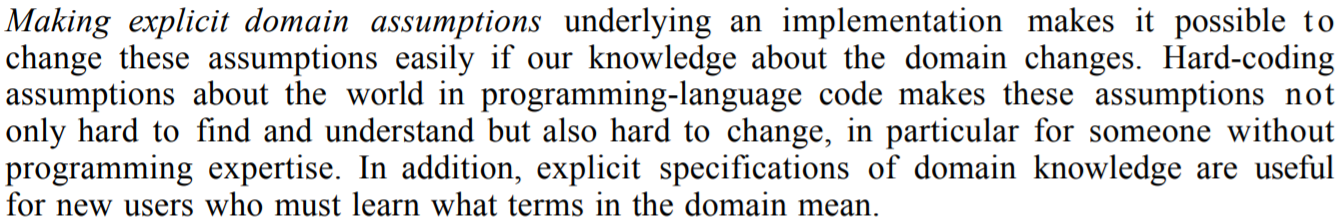


2，使知识可以重复利用【信息的分享】

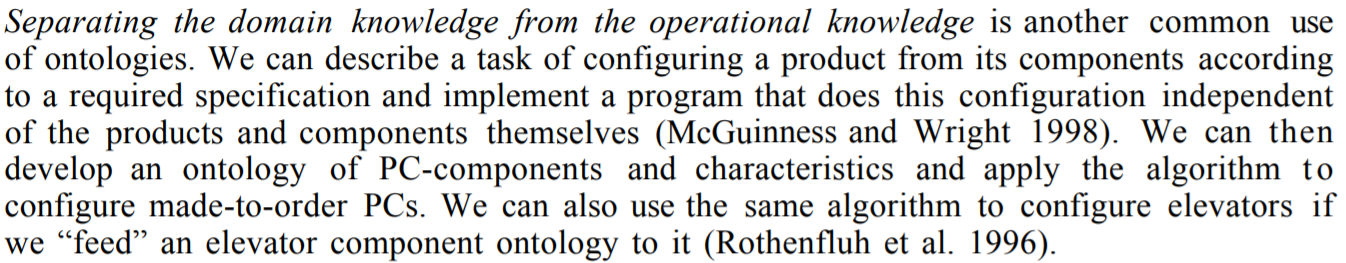




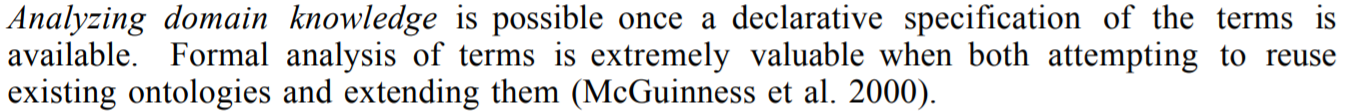
3，使领域的假设可以更加明确【知识的变化更加灵活，因此基于知识的假设的变化也更加灵活】



4，使领域知识和操作知识分离【信息之间低耦合】



5，使得知识可以被分析【数据挖掘】



因此，一般来说，不是为了设计开发本体而设计开发本体。设计开发本体的目的是通过定义数据集和结构，最终给程序来使用。

注意：本体开发和面向对象设计并不是同一概念。

# 本体是什么

ontology的定义：（维基百科）

In computer science and information science, an ontology encompasses a representation, formal naming, and definition of the categories, properties, and relations between the concepts, data, and entities that substantiate one, many, or all domains.

本体定义：

在某一特定领域内，本体由以下组成：被正式且明确描述的类（也称为概念）、用于描述每一个类内部的各个属性的关系（slot）以及用于描述类与类的关系（slot）、这种属性的限制条件。注意：定义中并不包含实体。

知识库=本体+各个类的实体集合。

类可以拥有子类，子类会比父类更加细化（从类的角度上）。

开发一个本体遵循以下步骤：

定义类；

将各个类归位为分类层级的形式；

定义slot，给这些slot描述好可取的值；

将值填入slots当中（并完成关系）

随后，只要定义好这些类的实体集，就可以创建一个知识库。

建议定义好类和slot的名字的命名规则，有外国人是这样定义的：类名每个单词都以大写字母开头，slot名每个单词以小写字母开头；

在图中，建议定义好类、实体的颜色。

Direct links表示slots；internal links有instance-of和subclass-of

# 创建本体的方法论

创建本体的方法还没有标准，以下引用斯坦福的方法：

这个方法的特点是有迭代性，首先先粗糙定义，不断修订和精炼本体，并填入细节。

本体设计中的基本规则：

1，对一个领域建立模型过程中，不存在一个绝对正确的方法。最好的方法总是取决于应用场景。

2，本体开发必然是一个迭代的过程（要通过实际应用检验、专家讨论）。

3，本体中的类（概览）要尽可能接近目标领域中的实体和关系。如果用一句话来描述特定的领域，那么本体一般就是那些名词，关系则是那些关系。

本体设计要直观性，可扩展性，可维护性，真实性（客观反映领域）。

第一步，决定本体的领域和范围

为了更好地设计你的本体，建议你问自己一些问题：

1，本体将会覆盖什么领域？

2，为了什么我们要用设计好的本体？

3，本体里的信息是用来回答哪些问题的？

4，谁将使用和维护本体？

第二步，考虑本体的重用性

第三步，枚举本体中的重要terms

第四步，定义类以及类的分级

第五步，定义类的属性slots

第六步，定义slos的限制条件

第七步，创建实例

# 本体和知识库的关系

本体是一种没有详细明确的、概念层面上的系统，用来支撑一个实际的知识库。

