

基于 Ontology 的领域知识构建技术综述

何海芸, 袁春风

(南京大学 计算机软件新技术国家重点实验室, 江苏 南京 210093)

摘要: 随着知识资源的不断丰富, 对知识进行有效的组织和管理, 实现知识共享和重利用, 从而充分有效地利用知识资源是一项迫切的研究课题。Ontology 的知识组织方式通过提供对领域知识的共同理解实现知识共享, 主要从定义、功能、构建、应用、研究现状等方面对 Ontology 进行了介绍分析, 并探讨了其研究方向和重点。

关键词: Ontology; 概念; 关系; 知识

中图法分类号: TP301; TP391

文献标识码: A

文章编号: 1001-3695(2005)03-0014-05

Overview of Technology of Building Domain Knowledge Based on Ontology

HE Hai-yun, YUAN Chun-feng

(State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing Jiangsu 210093, China)

Abstract: With the knowledge resources become more and more abundant, how to organize and manage knowledge efficiently in order to implement the share and reuse of knowledge and make use of knowledge more efficiently become a pressing research task. The construction of knowledge based on Ontology can implement the knowledge share by providing the common understanding of domain knowledge. In this paper, the definition, function, design and application about Ontology are introduced and analyzed. In the end, we discuss the research direction and emphasis in future are discussed.

Key words: Ontology; Concept; Relation; Knowledge

近年来, 随着计算机应用和网络技术的不断发展, 知识资源越来越丰富, 传统的一些知识处理和知识组织方式面对新的应用遇到了困难, 如传统知识库的构建方法已经很难适用于大型知识库的构建。由于知识表示的不统一, 知识存储的分布性等问题使得知识之间的互操作困难, 进而导致 Web 上丰富的资源得不到有效利用。如何对知识进行有效的组织和管理, 以便于知识的共享和重利用成为一项迫切而重要的研究课题。Ontology 是一种能在语义和知识层次上描述系统的概念模型, 其目的在于以一种通用的方式来获取领域中的知识, 提供对领域中的概念的共同一致的理解, 从而实现知识在不同的应用程序和组织之间的共享和重利用。

1 Ontology 简介

1.1 Ontology 的定义

Ontology 的发展覆盖的领域很多, 如哲学、人工智能领域、知识工程领域等, 在不同的领域有不同的定义, 关注的焦点也不同。Ontology 这个词来自于哲学, 在哲学中指的是对客观存在的系统的解释和说明。

在 AI 领域, Neches 等人将 Ontology 解释为“Ontology 定义了包含相关领域词汇的基本术语和关系, 以及组合这些术语和关系定义词汇外延的规则”^[1]。B. Chandrasekaran 等人的观点和 Neches 类似, 认为“Ontology 属于人工智能领域中的内容理论, 它研究特定领域知识的对象分类、对象属性和对象间的关系, 它为领域知识的描述提供术语”^[2]。

知识工程领域中把 Ontology 看成是一种工程制品 (Engi-

neering Artifact)。Gruber 给出“Ontology 是概念化的一个形式的规格说明”^[3]。所谓概念化可以理解为一组概念 (如实体、属性、过程) 及其定义和相互关系^[9]。Borst 在 Gruber 定义中加入了共享, 定义 Ontology 是被共享的概念化的一个形式化的规格说明^[4]。Studer 等人则进一步详细给出了 Ontology 的概念: “Ontology 是共享概念模型的明确的形式化规格说明”。其中, 概念模型指通过抽象出客观世界中一些现象的相关概念而得到的模型; 明确指所使用的概念以及使用这些概念的约束都有明确的定义; 形式化指 Ontology 是计算机可处理的; 共享指 Ontology 中体现的是共同认可的知识, 它不是对某个人而言的, 而必须是由一个团体承认的^[5]。William 等人从特征和形态方面对 Ontology 下定义, 认为“Ontology 用于描述或表达某一领域知识的一组概念或术语, 可用于组织知识库较高层次的抽象, 也可以用来描述特定领域的知识”^[6]。Fonseca 等人则从具体的建模元语来描述 Ontology, “Ontology 是一个理论, 它从特定的角度使用特定的词汇去描述实体、类、属性和相关的函数”^[8]。

本文关注的以及下面提到的都是指知识工程领域中的 Ontology。从上述内容可以看到, 目前对 Ontology 还没有一个统一的定义, 对 Ontology 定义的表述各不相同。虽然这些表述有差别, 但我们发现它们有一些共同的特征, 即概念化、形式化说明、领域知识和共同认可的客观存在。归纳之, Ontology 用于组织较高层次的知识抽象, 描述领域知识, 它通过捕获相关领域的知识, 确定领域内共同认可的概念和概念间的关系, 从而提供对该领域的共同理解。

1.2 Ontology 的分类

Ontology 的分类方式很多, 在文献 [25] 中给出了许多分类

的基准, 有 Ontology 的主题、Ontology 表示的形式化程度、Ontology 的目的、Ontology 是否在线、Ontology 是否共享等。Guarino 的观点则是主张从详细程度和领域依赖程度这两个维度对 Ontology 进行划分^[8]。

在具体应用中, 最常见的分类依据是依据 Ontology 的概念主题和形式化程度来划分。根据 Ontology 的概念主题一般可分为领域 Ontology、通用 Ontology、表示 Ontology、任务 Ontology^[24]; 根据 Ontology 的形式化程度分为高度非形式化 Ontology、非形式化 Ontology、半形式化 Ontology、严格形式化 Ontology^[9]。确定 Ontology 所属的类别, 明确 Ontology 的特征, 对 Ontology 构建过程中不同构建原则方法的选择具有很大的指导作用。

1.3 Ontology 的功能

总的来看, Ontology 可作为知识表达的基础, 避免重复的领域分析, 并通过统一的术语和概念达成知识共享的目的。

从具体的方面来看, 主要体现在通信 (Communication)、互操作 (Inter-operability) 和系统工程 (Systems Engineering) 上^[9]。通信方面, Ontology 提供共同的词汇使得人们和组织之间的交流准确无歧义。不同的模型方法、范例、语言和软件工具可以借助 Ontology 进行转换映射从而实现不同系统之间的互操作。在系统工程上主要体现在四个具体的方面, 从可重用性的角度来说, Ontology 能够对重要的实体、属性、过程及其相互关系进行形式化描述, 使得它们成为软件系统中可重用或者是共享的组件; 在知识获取上, 使用已有的 Ontology 作为基础来指导知识的获取, 能够提高获取的速度和可靠性; Ontology 提供的形式化表示可以自动对结果进行一致性检查, 使得软件系统更加可靠; Ontology 可以支持需求的识别以及能确定信息系统的规范^[9]。

1.4 Ontology 与其他一些数据知识组织的比较

传统的对数据或知识进行组织的方式有多种, 如数据库、辞典、百科全书、索引典、分类表和语义网络等。Ontology 类似于数据库模式, 两者都能在某种程度上独立于应用程序获得对数据或者知识的独立性。不同的是, Ontology 通过在应用程序之外去确定和管理领域的语义信息而获得语义的独立性, 而数据库模式通过建立规范及对应用程序之外存储的数据元素的管理获得数据的独立性。Ontology 提供的是领域的知识, 数据库模式则侧重的是提供数据容器的结构。Ontology 的目标重点在于知识的共享, 数据库模式则不侧重这个问题。在词典、百科全书等组织方式中, 它们按某种排序方式 (如字母顺序) 列出概念, 对概念的解释以及概念与其他概念之间的联系通过自然语言表达。Ontology 则使用形式化的方法将概念、概念的属性、概念之间的关系表达出来, 以方便计算机直接进行操作, 并能对知识进行推导。对于索引典而言, 虽然它也给出了概念之间的联系, 但是对于联系的类型并没有给出。可见, Ontology 在表述知识上是比较完整和全面的, 语义信息也很丰富, 而且它还提供了对知识推导的支持。

2 Ontology 的构建

Ontology 的构建是一个工程性的过程, 遵循一定的构建原则和构建方法学。但是目前业界没有一个统一的标准, 在建立

各自需要的 Ontology 时, 都是根据应用的具体情况并结合自身的经验使用自己的原则和方法来构建。以下是一些目前在知识工程界较为有名的遵循的构建原则和采用的构建方法学。

2.1 Ontology 的构建原则

Gruber 于 1995 年提出构建 Ontology 的五条原则^[10]:

(1) 明确性和客观性。Ontology 应能有效地说明所定义术语的内涵; 定义应该是客观的, 与背景独立的; 定义应该尽可能完整, 所有的定义应该用自然语言加以说明。

(2) 一致性。由 Ontology 推导出来的概念含义应该与 Ontology 中的概念含义本身一致。Ontology 所定义的公理以及用自然语言说明的文档都应该是一致的。

(3) 可扩展性。添加新的概念时, 不需修改原有内容。

(4) 最小编码偏差。Ontology 不应该依赖于某一特殊化的符号层的表示方法。

(5) 最小 Ontology 承诺。Ontology 的承诺应该最小, 只要能满足特定的知识共享需求即可, 即对建模对象给出尽可能少的约束。

此后, 其他人还陆续补充了一些原则:

(1) Ontology 区分原则。Ontology 中的类之间是不相交的^[11]。

(2) 使用多种概念层次, 多重继承机制来增加表达能力^[13]。

(3) 最小化同层相邻概念之间的语义距离^[13]。

由于没有统一的原则, 而且各人实际构建的 Ontology 的类型和应用情况也不同, 这时需要根据客观实际选择合适的原则, 并在具体工作中进一步细化这些原则。

2.2 构建 Ontology 方法学

目前, Ontology 工程还处于一种不成熟的阶段, 很多 Ontology 工程都使用自己的方法学来开发各自的 Ontology, 比较有名的有以下几种:

(1) Mike Uschold 和 King 的“骨架”法^[14,24]。这个 Ontology 建立模式是爱丁堡大学从开发 Enterprise Ontology 的经验中产生, 过程如下: 根据任务情况确定 Ontology 的应用目的和范围, 明确需求之后进入 Ontology 构造阶段。这一阶段包括首先对 Ontology 进行分析, 确定 Ontology 中的概念和关系, 然后选择合适的语言来形式化这些概念和关系, 最后对这些概念和关系进行集成, 使之成为整体。构建阶段之后是 Ontology 的评价阶段, 这一阶段要根据第一阶段中确定的需求和 Ontology 的能力问题对 Ontology 以及软件环境、相关文档进行评价, 如果符合要求, 则最后形成文档, 如果不符合, 则需重新回到构造阶段的本体分析。其流程图如图 1 所示。

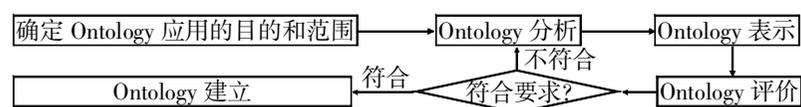


图1 “骨架”法流程图

(2) Gruninger 和 Fox 的“评价法”^[14]。这个方法是根据多伦多大学的 TOVE (TOronto Virtual Enterprise) 项目, 由 Gruninger 和 Fox 等人提出的。其构建过程由五个阶段组成: 收集应用情景阶段, 根据收集的应用情景提供问题的解决方案, 并依据这些方案确定所要定义的概念和关系的内在语义。非形式化 Ontology 能力问题的形成阶段, 这些问题是构造 On-

tology 的约束以及对 Ontology 的评价标准。术语的抽取和定义阶段, 即从问题中抽取出涉及到的词汇, 然后进行形式化定义。词汇形式化之后, 在下一个阶段就能将问题也形式化定义。对 Ontology 中的词汇公理进行定义, 主要用一阶谓词逻辑表示, 公理主要是用来定义词汇的语义和约束用的。其流程如图 2 所示。

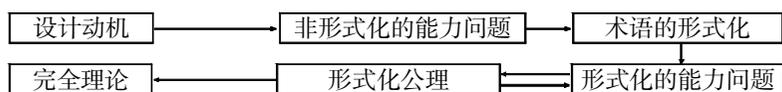


图 2 TOVE 流程图

(3) Methontology 方法。该方法是由西班牙马德里理工大学人工智能实验室提出的, 其特色在于提出用生命周期的概念来管理整个 Ontology 的开发过程, 使 Ontology 开发过程更接近软件工程开发方法。它分为三个不同的阶段: 管理阶段。主要是对任务进行系统规划, 规划内容包括进展情况、资源情况以及质量保证问题等。开发阶段。通过规范说明、概念化、形式化、实现和维护等步骤对 Ontology 进行开发。维护阶段。包括知识获取、系统集成、评价、文档说明、配置管理等。

这些方法学都是从各自的开发过程中通过实践经验总结出来的, 需求分析、构建和评价是这些方法学中共同的阶段, 它们都在一定程度上接近软件工程的开发过程。但文献[14]中将上述方法同 IEEE 标准软件开发生命周期法 IEEE1074-1995 进行了比较分析, 指出这些方法都没有完全成熟。目前知识工程界缺少公认的成熟的 Ontology 建模标准以及开发指导原则和可操作性方法, 因此在一定程度上影响着 Ontology 的重用和共享。这成为目前的研究方向和重点。

2.3 相关技术

2.3.1 描述语言

Ontology 的表示方式多种多样, 可以用自然语言、框架和逻辑语言等来描述 Ontology。自然语言是一种非形式化的表示方法, 一般用于在构建的初期或者文档中进行描述; 框架表示法使用框架将概念、概念的属性、概念之间的关系清晰地表示出来, 框架代表概念, 框架的槽可以描述概念的属性, 也可以表示与其他概念的关系, 其语法是高阶的, 框架的槽值可以是另一个框架; 逻辑语言主要使用谓词逻辑语言对 Ontology 进行描述。框架的描述方式比较自然、形象, 但是其本身的推导能力很弱, 逻辑语言的推导机制则较强。由于这一点, 一些描述 Ontology 的语言综合了逻辑和框架语言的特性。目前已经出现了许多描述 Ontology 的语言, 比较有名的有以下几种:

(1) Ontolingua。它是一种基于 KIF 提供的统一的规范格式来构建 Ontology 的语言, KIF(Knowledge Interchange Format) 是一种用于不同计算机系统之间交换知识的一阶语言。Ontolingua 提供了对类、关系、函数、对象和公理进行定义的表示形式, 它使用 Frame Ontology 支持二阶关系的表示。由 Ontolingua 构造的 Ontology 可以很方便地转换到各种知识表示和推理系统, 能很好地适应于不同的系统以及进行系统之间的移植^[3,15]。

(2) CycL。它是 Cyc 系统的描述语言, 具有一阶谓词演算能力, 并扩充了等价推理、缺省推理等功能, 同时还扩展了一些二阶谓词演算的特性。

(3) OIL(Ontology Inference Layer)。它综合了三个方面的

特性: 描述逻辑(Description Logic)提供的形式语义和有效的推导支持、框架提供的建模元语和 Web 提供的为句法交换标记所提供的标准。因此, OIL 能为开发者提供大量的基于框架的 Ontology 的建模元语, 对 Ontology 的描述具有描述逻辑的简单、清晰、良定义的语义, 并能提供进行自动一致性检测和包含性确认。

(4) OWL(Web Ontology)。OWL 语言是用来定义和实例化 Web 上的 Ontology 的, 它使用 RDF/XML 的语法形式定义。一个 OWL Ontology 包含描述类、属性以及它们的实例。OWL 定义的 Ontology 之间是相互联系的, 一个 Ontology 能够显式引用其他的 Ontology 或 Ontology 中的信息。

不同的语言其特点不同, 各有其适用之处, 在具体使用时要根据具体应用选择合适的语言。如 Ontolingua 利用其易于转换的特性, 更适合在知识层次上表示知识, 具体的实现可以通过它进行转换, 这使得对 Ontology 的维护与使用它的目标表示系统分离开来。OIL 和 OWL 与 Web 上一些标准表示相结合的特点, 使得它们主要用于 Web 上的 Ontology 的表示。

2.3.2 辅助构建工具

为了便利 Ontology 的开发, 出现了很多 Ontology 的编辑工具, 如 Ontolingua Server^[16,17], Java Ontology Editor^[17,26], OntoEdit^[17], OilEd^[17,27], Protege^[17,28], Chimaera^[17,29] 等。它们能很好地辅助进行 Ontology 的编辑、修改、浏览和维护等工作。

(1) Ontolingua Server。它是有代表性的协作式的 Ontology 建造工具, 用于辅助 Ontology 的协作式开发。可以对 Ontology 进行浏览、创建、编辑、修改和使用, 而且它可以通过 Web 来发表、浏览、创立和编辑存储在 Ontolingua Server 上的 Ontology。

(2) OntoEdit。它是一个 Ontology 工程环境, 集合了基于方法学的 Ontology 开发以及协调和推导的能力。从方法学角度来考虑, OntoEdit 主要注重于 Ontology 开发中的三个步骤: 需求规范、精化和评价。

(3) Chimaera。它是一个基于 Web 的 Ontology 浏览的环境, 它接受超过 15 种指定的输入形式的选择, 如 KIF, Ontolingua, Protégé 和 CLASSIC 等, 同时还有其他的 OKBC 兼容的形式。它提供的两个主要功能是: 将多个 Ontology 合并起来, 对单个或者多个 Ontology 进行诊断。

2.3.3 半自动、自动构建技术

AI 领域中许多机器学习的方法被改进应用到 Ontology 的学习中, 实现 Ontology 的半自动或自动构建。

机器学习的类型有在线学习、离线学习、指导性学习和非指导性学习, 不同的机器学习类型适用于不同类型的 Ontology 构建。离线学习适用于那些更改速度较慢的 Ontology, 在线学习反之; 指导性学习适用于规模较小的 Ontology, 非指导性学习反之。如领域的 Ontology 是属于更新速度较慢规模也较小的, 可以借鉴离线指导性的机器学习方法进行构建^[18]。

机器学习中的许多技术可以被 Ontology 借鉴利用。借助聚簇技术可以帮助解决概念的分类问题; 增量式的学习方法可以应用到 Web 上的 Ontology 的更新中; 关联挖掘技术可以帮助从语料库、文本中发现概念, 丰富已有的 Ontology 中的概念, 关联规则的发现还可以帮助提取概念之间的层次性关系及确定适当的抽象层次等。

将机器学习直接应用于 Ontology 学习的一个最大障碍是: Ontology 最终是人们可理解的, 然而机器学习的结果是非理解的形式。所以需要机器学习技术进行一定的修改, 使其适用于 Ontology 的学习^[18]。

将机器学习应用到 Ontology 学习中, 实现 Ontology 的半自动、自动构建能在很大程度上加快 Ontology 的构建进程, 节省很多的人力和时间。Ontology 的学习(OL) 目前也成为研究的重点之一。

3 基于 Ontology 的应用

Ontology 的功能主要是实现某种程度的知识共享和重用, 它能使得计算机对信息和对语言的理解上升到语义层次。所以, Ontology 在一些涉及到信息的互操作、知识理解等方面的领域具有很大的应用前景。

在信息检索方面, Ontology 使得传统的基于关键字的检索, 上升到语义检索的高度。其基本思想是: 先建立相关领域的 Ontology, 根据 Ontology 对收集的信息进行标注, 用户的检索请求按照 Ontology 转换成规定的格式, 在 Ontology 的帮助下匹配出符合条件的数据集返回给用户。目前 Ontology 应用在信息检索中的著名项目包括 (Onto)² Agent, Ontobroker 等。(Onto)² Agent 的目的是为了帮助用户检索到所需要的 WWW 上已有的 Ontology, Ontobroker 面向的是 WWW 上的网页资源, 目的是为用户检索到所需要的网页^[19]。

在信息集成方面, 分布式信息集成面临的主要问题是结构、设施的异构和缺乏统一的语义集, 借助 Ontology 可以在一定程度上解决语义异构的问题。采用语义方法进行信息集成的特点是扩展性好、适应动态信息源、支持语义级查询。集成方式有两种: 自底向上, 自顶向下。自顶向下方式的基本思想是先建立相关领域的 Ontology, 然后由该 Ontology 来统一底层各信息源的语义。自底向上方法是先提取底层各信息源的局部数据模式, 再在局部数据模式上抽取局部概念模式, 最后在局部概念模式上构造全局概念模式^[20]。

在机器翻译方面, 通过把某种语言中的词汇映射到 Ontology 中的概念, 可以支持在源语言分析时进行歧义消解和目标语言生成时的词汇选择, 并可以作为源语言和目标语言的中间表示的概念来源^[21]。现在已经建立的一些共享的 Ontology (如 CYC, Wordnet, Sensus 等) 都能帮助自然语言理解。

另外, 在知识获取方面, 借助 Ontology 能够更加有效地获取知识; 在数据挖掘中, 基于 Ontology 的数据挖掘可在高层次进行, 产生高层次或多层次的规则, 甚至在具有语义意义的规则上产生挖掘结果; 在软件工程方面, Ontology 能帮助更加准确地获取软件需求信息。这方面, 国内已有相应的研究正在开展, 还有在多 Agent 系统的自动设计、B2B 电子商务和 CSCW 等方面都引入了 Ontology。

由于 Ontology 构建的困难, 以及构建技术不成熟等原因, 现在真正能对 Ontology 进行的应用还很少。在各个应用点方面也只是一种设想或者是在小型的 Ontology 上进行实验。大型的的实际的应用仍然依赖于构建技术的突破。

4 Ontology 的研究与应用现状

Ontology 自 20 世纪 90 年代提出以来, 在国外受到了普遍

的关注, 相关方面的研究已经进行, 很多大型的工程也在开展。国内对这方面的研究近几年来也在逐步地开展, 但相对而言还比较少。

在理论研究方面, 比较有代表性的是 Guarino 等人对概念及其分类开展的研究工作^[19]。他们对概念的分类进行了深入细致的研究, 从一般的意义上对概念的定义、概念的特性、概念之间的关系及概念的分类进行了分析, 提出了一套指导概念分类的可行理论。他们还进一步提出了 Ontology 驱动的建模方法, 从而在理论上为建模提供了一个通用的模式。

在实际应用方面, 许多工程在开展之中, 比较有名的有: CYC^[9], Ontosaurus^[9], Enterprise^[9], TOVE^[22], KACTUS^[9], PLINIUS^[9], PANGLOSS^[23], Loom^[17], OntoWeb^[17], Ontoknowledge^[17] 等。

4.1 CYC

CYC 是位于美国德州奥斯汀的 MCC (Microelectronics and Computer Technology Corporation) 公司的研究项目。它的主要目的是通过开发广泛的基于特定领域应用的 Ontology 来提供常识推导的基础。所有的 CYC 中的知识是用 CYCL 以断言的形式展示和声明的。CYCL 知识库本身包含一些简单的断言、推导规则和用于推导的控制规则; 它的推导引擎能用来从该知识库中推导出新的断言。CYC Ontology 按照模块 (Module) 组织, 称为微理论 (Microtheories)。每个微理论包括某一特定领域知识和推理所需的概念。在一个领域中可能存在多个微理论, 反映了人们对该领域的多个不同角度的看法。

4.2 KACTUS

KACTUS 是欧洲的 ESPRIT 项目。其目的是开发出技术系统全生命周期的知识重用方法学, 从而能在设计、诊断、操作、维护、再设计和培训时使用同一知识库。KACTUS 通过构建 Ontology 以及在不同的应用中重利用它们来实现这一点。

在 KACTUS 中, 主要的知识表示形式是 CML (Conceptual Modeling Language), KACTUS 提供了一个交互式的工具来浏览、编辑和管理 Ontology。KACTUS 支持面向应用和理论的工作包, 它提供了一个既能对理论问题做实验, 又能进行实际工作的环境。

KACTUS 工具能提供对 Express 和 Ontolingua 的支持。

4.3 PANGLOSS

PANGLOSS 工程主要是为了构建一个大规模的基于知识的机器翻译系统。其中一个关键的部分是构造 PANGLOSS Ontology。该 Ontology 主要是用于支持其他的 PANGLOSS 模块进行语义的处理, 它是通过对不同的在线字典、语义网络和双语资源进行组合半自动构建起来的, 其中包括 Penman Upper Model, Ontos, LDOCE, WORDNet 和 Collins Bilingual Dictionary。该 Ontology 包括三个层次: 最高层 Ontology Base (OB), 由大约 400 个有效地涵盖不同的 PANGLOSS 模块所需的一般性、概括性的节点构成; 中间层大约包含 50 000 个节点, 提供了一个普通世界模型的框架, 用英语词汇的意义来表示; 最底层提供了对不同应用领域的描述。

4.4 Ontoknowledge

Ontoknowledge 是一个正在进行的第五代欧洲框架程序工

程, 它提供对数字网络上的信息进行访问, 支持有效的知识管理。它的主要目的是提供对弱结构化的在线信息资源进行访问、获取和维护。Ontoknowledge 开发了一个三层结构来提供对信息的访问: 在最底层(信息层), 弱结构信息资源经过处理抽取出机器可处理的元信息; 中间层(表示层) 使用这些元信息提供对信息资源的自动访问、创建和维护; 最高层(访问层) 使用基于 Agent 的技术同时还有人工查询技术和可视化技术来指导用户去访问这些信息。

在国内, 中国科学院建立了一定规模的 Ontology, 并应用到自动问答系统中; 天鹰软件工程项目中也使用 Ontology 对知识进行组织和管理。不过相对而言, 国内在这方面开展的大型工程还是比较少的。

5 结束语

Ontology 作为一种新的知识组织方式, 力图去解决知识的共享和重用问题, 在知识越来越丰富的今天, 受到了越来越多的关注, 在许多方面有着广泛的应用前景, 许多研究也都相继开展起来。然而, 我们也看到, 基于 Ontology 知识库系统理论及应用还处于初步阶段, 其理论和方法还有待于进一步完善。其主要困难体现在以下几个方面:

(1) 目前的 Ontology 很多都是人工开发的, 这样需要耗费很多的人力、物力和财力, 时间周期也很长, 这在一定程度上影响了 Ontology 的应用。

(2) Ontology 构建的原则、方法及其表示等许多方面都没有形成一个统一的标准, 这也使得 Ontology 只是作为某一个单独的团体或组织内的共享, 真正意义上的共享和重用仍然没有实现。

(3) Ontology 的集成方法目前也不成熟。

(4) 在 Ontology 的理论基础方面, Ontology 的评价方法以及形式化方法还需要进一步研究与探讨。

今后 Ontology 的研究重点也将围绕着这些困难继续进行, 在理论与应用两个方面不断地深入下去。

参考文献:

- [1] Neches, R E Fikes, T Finin, *et al.* Enabling Technology for Knowledge Sharing [J]. *AI Magazine*, 1991, 12(3): 36-56.
- [2] B Chandrasekaran, *et al.* What are Ontologies, and Why do We Need Them [J]. *IEEE Intelligent Systems*, 1999, 14(1): 20-26.
- [3] Thomas R Gruber. Ontolingua: A Translation Approach to Potable Ontology Specification [J]. *Knowledge Acquisition*, 1993, 5(2): 199-200.
- [4] Borst W N. Construction of Engineering Ontologies for Knowledge Sharing and Reuse [D]. PhD Thesis, Enschede: University of Twente, 1997.
- [5] Rudi Studer, V Richard Benjamins, Dieter Fensel. Knowledge Engineering: Principles and Methods [J]. *IEEE Transactions on Data and Knowledge Engineering*, 1998, 25: 161-197.
- [6] William S, Austin T. Ontologies [J]. *IEEE Intelligent Systems*, 1999 (1/2): 18-19.
- [7] Perez A G, Benjamins V R. Overview of Knowledge Sharing and Reuse Component: Ontologies and Problem-Solving Methods [C]. *Proceedings of the IJCAI-99 Workshop on Ontologies and Problem-solving Methods (KRR5)*, 1999. 1-15.
- [8] Nicola Guarino. Formal Ontology and Information Systems [C]. *Proceedings of FOIS '98*, 1998. 3-17.
- [9] Mike Uschold, *et al.* Ontologies: Principles, Methods and Applications [J]. *Knowledge Engineering Review*, 1996, 11: 93-155.
- [10] Gruber T R. Towards Principles for the Design of Ontologies Used for Knowledge Sharing [J]. *International Journal of Human-computer Studies*, 1995, 43: 907-928.
- [11] A Gómez-Pérez. Knowledge Sharing and Reuse [R]. *Handbook on Applied Expert Systems*, 1998.
- [12] A Bemaras, I Laresgoiti, J Corera. Building and Reusing Ontologies for Electrical Network Applications [C]. *Proceedings of the 12th ECAI*, 1996. 298-302.
- [13] J Arpirez, A Gómez-Pérez, *et al.* (Onto)² Agent: An Ontology-based WWW Broker to Select Ontologies [C]. *ECAI*, 1998. 16-24.
- [14] 杨秋芬, 陈跃新. Ontology 方法学综述 [J]. *计算机与信息技术*, 2001, (10): 2-6
- [15] Thomas R. Gruber Ontolingua: A Mechanism to Support Portable Ontologies [R]. *Technical Report*, Knowledge Systems Laboratory.
- [16] Adam Farquhar, Richard Fikes, James Rice. The Ontolingua Server: A Tool for Collaborative Ontology Construction [R]. *Technical Report*, Stanford KSL, 1996. 26-96.
- [17] Mar á Auxilio, Medina Nieto. An Overview of Ontologies [R]. *Ex. Hacienda Sta. Catarina Mártir s/n Cholula*, 2003.
- [18] Borys Omelayenko. Machine Learning for Ontology Learning [R]. *Amsterdam the Netherlands: Report for the Course IT3 Knowledge Acquisition and Sharing at the International Jyvaskyla Summer School*, 2000.
- [19] 邓志鸿, 唐世渭, 张铭, 等. Ontology 研究综述 [J]. *北京大学学报 (自然科学报)*, 2002, 38(9): 728-730.
- [20] 邓志鸿, 唐世渭, 杨冬青. 面向语义集成——本体在 Web 信息集成中的研究进展 [J]. *计算机应用*, 2002, 22(1): 15-17.
- [21] 王小捷, 钟义信. 基于 Ontology 的英汉机器翻译研究 [J]. *中文信息学报*, 2000, 14(5): 8-15.
- [22] 王昕. 综述: 本体的概念、方法和应用 [EB/OL]. <http://www.prdm.net/papers/knowledge/Ontology%20overview.htm>
- [23] Kevin Knight, *et al.* Building a Large Scale Knowledge Base for Machine Translation [C]. *Proc. of AAAI*, Seattle, 1994. 773-778.
- [24] 王晓东. 基于 Ontology 知识库系统建模与应用研究 [D]. 上海: 华东师范大学.
- [25] 阮明淑, 温达茂. Ontology 应用与知识组织之初探 [J]. *佛教图书馆馆讯*, 1991, (32): 6-17.
- [26] JOE [EB/OL]. <http://www.cse.sc.edu/research/cit/demos/joe/>.
- [27] Bechhofer S, Ian Horrocks, Carole Goble, *et al.* Oiled: A Reasonable Ontology Editor for the Semantic Web [R]. *Lecture Notes in Computer Science*, 2001.
- [28] Noy N F, McGuinness D L. Ontology Development 101: A Guide to Creating Your First Ontology [R]. *Stanford Knowledge Systems Laboratory Technical Report KSL-01-05 and Stanford Medical Informatics Technical Report*, 2001.
- [29] Chimaera [EB/OL]. <http://www.ksl.stanford.edu/software/chimaera>.

作者简介:

何海芸 (1979-), 女, 硕士研究生, 主要研究方向为知识处理、自然语言理解; 袁春风 (1962-), 女, 教授, 主要研究方向为知识处理、多媒体信息处理、自然语言理解等。