

模糊语义网的体系结构及其推理研究

学 科: 计算机软件与理论

研究生签字:



指导教师签字:



摘 要

语义网这个概念于 2000 年首次由 Berners-Lee 提出, 以往 Web 技术中计算机主要扮演展现信息的角色, 几乎不参与信息处理, 忽略计算机信息处理的作用, 一方面使得 Web 中庞大数据无法得到有效使用, 另一方面造成智能应用无法在 Web 上展开。语义网的目标是让机器理解 Web 上的信息, 来应对快速增长的 Web 资源, 为人类提供更好的服务。目前的语义网, 将资源信息和资源间关系看成固定的。而我们知道, 信息的语义在不同的社会环境、不同文化、不同的领域中是不同的; 同时, 信息的语义还会随着外界因素的变化而发生演变。所以, 我们很有必要扩充语义网理论, 从语义网体系结构的资源描述层入手, 引入描述资源的模糊性、资源间关系的模糊性及其发展变化的成分。因此, 本文在研究现有语义网理论的基础上, 研究了模糊语义网的概念, 初步给出了模糊语义网的体系结构。并重点研究了模糊语义网中与语义紧密相关的具体关系描述层和抽象关系描述层, 给出了具体关系和抽象关系的描述方法及其演化机制! 同时给出了模糊语义网的演化算法, 并设计了职称晋升本体及其演化程序来进行验证。

本文具体做了以下研究工作:

- 1) 学习了语义网的思想及现状。
- 2) 重点研究了语义网的体系结构中的 RDF(S)和本体层, 相应的添加了模糊成分。初步提出模糊语义网的体系结构。
- 3) 学习了模糊认知图原理, 将其原理应用到资源描述框架 RDF(S)中, 在其三元组中的属性添加了权值, 形成了模糊资源描述框架 FRDF, 提高了复杂知识的表示与处理能力。
- 4) 学习了语义网的本体层, 给出了模糊本体的定义及其形式化描述, 并给出了抽象层次上的模糊本体 Web 语言 FOWL 的语法, 用 FOWL 描述模糊本体。
- 5) 研究了模糊语义网的演化, 学习了神经网络, 把反馈神经网络的原理用于模糊语义网的演化中, 提出了具体关系的演化算法。同时也给出了个体隶属度计算方法。
- 6) 设计了职称晋升本体及其演化的实验, 验证了模糊语义网理论的可行性。

关键词: 语义网; 模糊本体; FOWL; 演化; 具体关系; 抽象关系; 模糊语义网

Research on the Architecture of Fuzzy Semantic Web and Reasoning on It

Discipline: Computer Software and Theory

Student Signature: Wu Yuyun

Supervisor Signature: Luo Junmin

Abstract

The concept of the Semantic Web was advanced by Berners-Lee in 2000. Computer in the former web showed information only, seldom involved in information processing. The former web ignored computer's use in information processing. The former web made the vast data cannot be effectively use and caused intelligent applications can not be used on the web. The target of the Semantic Web was that information on the web can be comprehended by computer. Accordingly, information on the web can be handled spontaneously to accommodate the quick growth of resources on the web and provided better service to human. At present, the theories of the Semantic Web put information and relations of resources as definitive. However, the semantic meaning of information is different among distinct domain, culture and social environment. Meanwhile, the semantic meaning also evolutes along with the change of the external world factor. So expanding the theory of the Semantic Web and pulling in ambiguity of resource description, relations among resources and changes of that are needed. Therefore, the thesis researched existing theory of the Semantic Web. On this basis, the thesis put forward the architecture of fuzzy Semantic Web and mainly studied on the concrete and abstract relationship description layers that are related to semantic and gave evolution machinery and methods to describe concrete and abstract relationship. At the same time, the thesis gave evolutionary algorithm of the Fuzzy Semantic Web and did analog experiment.

The thesis researched the following research work specifically:

- 1) Researched the mind and development status of the Semantic Web.
- 2) Studied on layers RDF(S) and ontology in the architecture of fuzzy Semantic Web mainly and pulling in ambiguity of resource description. On this basis, the thesis put forward the architecture of fuzzy Semantic Web.
- 3) Discussed the theory of the Fuzzy Cognitive Map. Applied the theory of the Fuzzy Cognitive Map to RDF(S). The thesis applied the theory of the Fuzzy Cognitive Map to RDF(S). The thesis added weight to property of the Resource Description Framework triples to express ambiguity and formed the Fuzzy Resource Description Framework. So as to enhance the ability

to express and deal with complex knowledge and semantic.

4) Studied ontology layer of semantic web, the thesis gave the definition of fuzzy ontology and its description, the thesis gave fuzzy OWL and use it to describe fuzzy ontology.

5) Studied evolution of the Fuzzy Semantic Web. The thesis applied the theory of the Feedback Neural Network to evolution of the Fuzzy Semantic Web and proposed evolutionary algorithm of the Fuzzy Semantic Web.

6) Constructed the analog system of the Fuzzy Semantic Web and evolution preliminarily on the basis of theoretical research. In order to test and verify application prospects of growing Fuzzy Semantic Web on enhancing intelligent of information search.

Key Words: semantic web; fuzzy ontology; fuzzy OWL; evolution; concrete relationship; abstract relationship; fuzzy semantic web

目 录

1 绪论	1
1.1 问题的提出及研究意义	1
1.1.1 问题的提出	1
1.1.2 研究意义	3
1.2 国内外研究现状	3
1.3 本文的主要内容和结构安排	4
1.3.1 本文的主要内容	4
1.3.2 本文的章节安排	5
2 模糊语义网的体系结构	6
2.1 语义网概述	6
2.1.1 语义网的概念	6
2.1.2 语义网的特征	6
2.2 语义网的架构	6
2.3 语义网的研究问题	7
2.4 模糊语义网的提出	8
2.5 模糊语义网的体系结构	8
2.6 模糊语义网的研究问题	10
3 具体关系及其描述	12
3.1 模糊认知图	12
3.2 资源描述框架 RDF	14
3.2.1 RDF 的含义	14
3.2.2 RDF 模型	14
3.2.3 RDF/XML 语法	15
3.3 模糊资源描述框架 FRDF	16
3.3.1 模糊资源描述框架 FRDF	16
3.3.2 利用模糊认知图表示 FRDF	17
4 抽象关系及其描述	19
4.1 本体	19
4.1.1 本体的定义	19
4.1.2 本体语言 OWL	20
4.1.3 OWL 语法	20
4.2 模糊本体	23
4.2.1 模糊本体及其形式化	23
4.2.2 模糊本体 Web 语言-FOWL	24

4.2.3 用 FOWL 表示模糊本体 27

5 模糊语义网的演化研究 30

5.1 具体关系的演化研究 30

5.1.1 反馈神经网络 30

5.1.2 具体关系的演化算法 31

5.2 抽象关系的演化研究 33

5.2.1 抽象关系的演化算法 33

5.2.2 个体隶属度的计算方法 35

6 职称晋升本体及其演化实验 45

6.1 实验系统的设计及实现 45

6.1.1 系统功能分析 45

6.1.2 系统结构的设计 45

6.1.3 系统流程 46

6.1.4 系统实现步骤 47

6.2 系统的开发平台和工具介绍 50

6.2.1 开发平台 50

6.2.2 开发工具 50

6.3 实验结果分析 51

7 结论 56

7.1 总结 56

7.2 展望 57

参考文献 59

攻读硕士学位期间发表的论文 61

致谢 62

学位论文知识产权声明 63

学位论文独创性声明 64

1 绪 论

Web 是互联网上的上层应用, 尽管诞生仅十多年, 但极大的关注和频繁使用促进了这项技术的发展。从静态 HTML 页面, 到动态 Web 页面; 从对内容的简单描述, 到将内容的结构和表示分离处理, 无论它的功能还是技术都变化了。到上世纪 90 年代, 随着 Web 信息的极速膨胀, 人类逐渐意识到缺乏自动处理大量 Web 信息的技术。鉴于此, Berners-Lee 在 2000 年提出了语义网的概念。它改变了计算机在 Web 中扮演的角色, 使得 Web 不但可以保存和重现信息, 而且能对信息进行智能化的处理。

1.1 问题的提出及研究意义

1.1.1 问题的提出

传播和记录知识使人类文明得以延续和发展。从远古结绳记事到如今大容量存储设备和光纤通信设备, 知识比任何时候都更容易被保存和传播。在信息时代的今天, Web 极大改变了人类利用知识的形式。一方面作为信息载体, 以丰富的表现形式展现信息; 另一方面提供多种检索、查询服务, 使人们方便获取信息。Web 的高速发展使它很快成为一个庞大的知识库, 但同时带来了许多问题。

1) Web 信息无法被自动的处理。不论手写的 HTML 网页, 还是查询时动态生成的网页, 目的都是让人阅读, 计算机只显示信息, 没有理解和处理的能力。因此, 在 Web 上开发各种智能应用是比较困难的。

2) Web 信息无法被有效的使用。如今随着 Web 规模的日益庞大, 使得对 Web 信息的有效利用成为了巨大的挑战。传统搜索引擎已无法应对 Web 这个日益庞大的知识库。拿 Google 来说, 目前搜索到 80 多亿的 Web 页面, 这仅仅占整个 Web 的 25%~30%, 大量的 Web 信息无法被搜索^[1]。同时, 由于计算机无法精确识别 Web 上的内容, 当前搜索引擎返回的信息要么过少, 要么过多, 搜索结果的质量并不令人满意。

3) Web 服务的异构问题。Web 服务方面, 尽管有 WSDL、UDDI 等技术标准, 仍存在诸多尚待解决的问题: 服务匹配、发现、检索的查准率较低; 服务集成需要人工干预, 无法自动化。不论是简单网页信息, 还是网络服务等, 主要问题均在于: 语义的二义性使得机器无法自动理解和处理它们。

例如, 如图 1.1 所示, 在搜索引擎中输入“苹果”一词, 第一页返回 90% 的信息都是数码方面的, 而仅有一条作为水果的“苹果”信息, 以“苹果”为品牌的服装和电影《苹果》等信息却没有显示。也就是说, 当前的搜索引擎在检索时只考虑了部分用户的偏好情况, 不全面。此外, 先前用户输入的“雪莲果”, 随后这一用户又输入“苹果”。根据前一次用户输入, 可以推断出, 此次用户的偏好是水果“苹果”, 而不是数码或是其他

方面的信息,水果“苹果”的可能性要大一些。但是,此次返回的信息仍和图 1.1 完全一样。

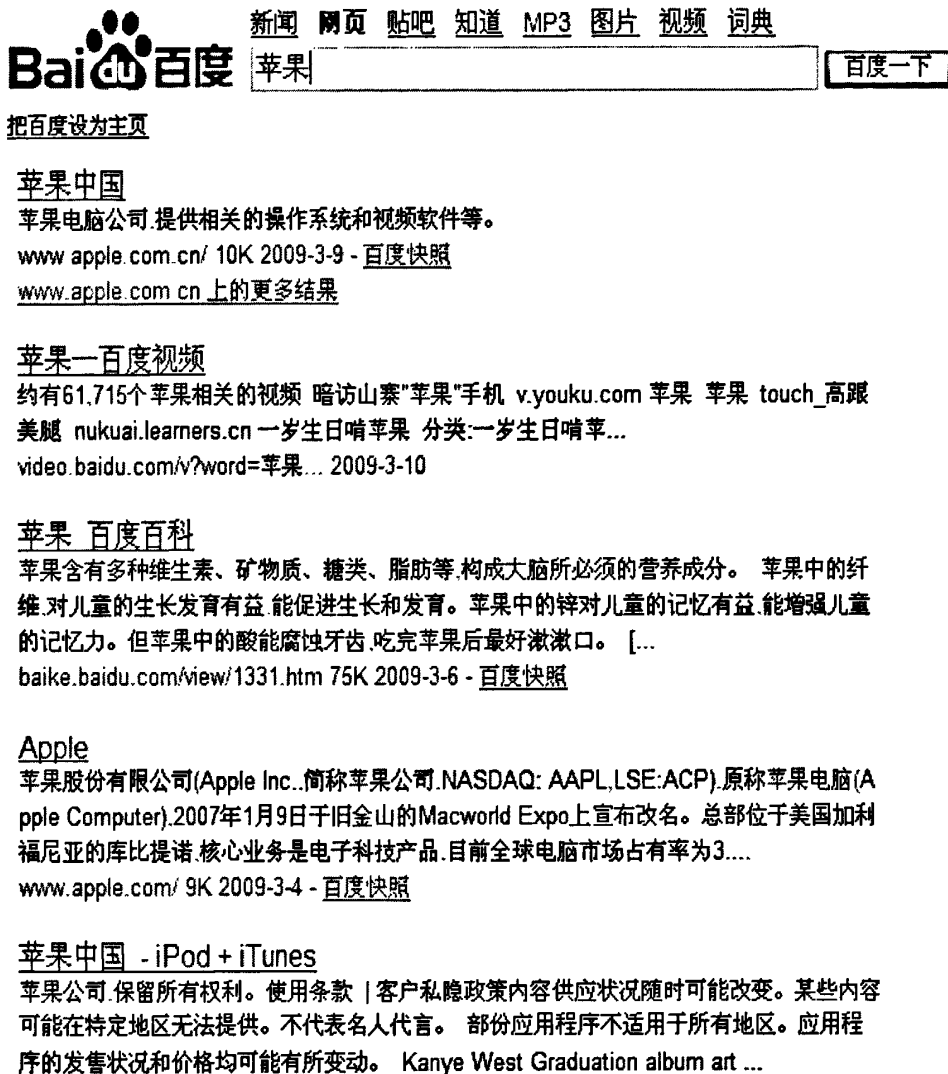


图 1.1 检索示例

针对上述问题, Berners-Lee 于 2000 年提出了下一代 Internet 的概念——语义网, 并首次给出语义网的体系结构, 指明了研究框架。语义网中的层次关系——以 XML 和 RDF/RDFS 为基础, 在此基础之上构建本体以及逻辑推理规则, 用来完成语义的知识表示与推理, 从而能被计算机理解 and 处理, 更有效的利用资源、更好的为人类服务。

目前, RDF 数据模型将资源间的关系描述为确定不变的, 描述的都是确定不变的信息。但是, 现实世界中存在很多模糊信息, 同时信息之间的关系也有模糊性, 即 RDF 数据模型无法描述不确定性信息。这就要求 RDF 数据模型将 Web 上资源本身的模糊性及资源间关系的模糊性表示出来。在此基础上的逻辑推理能够更接近人的思维, 更逼真的描述

现实社会,增强计算机的智能性。而且,信息的语义在不同的社会环境,不同的文化、不同的领域中是不同的。同时,信息的语义还会随着外界因素的发生演变。那么,模糊语义网的体系结构是怎样的?其关键层 RDF(S)和本体层中是如何引入模糊权值的?具体关系及抽象关系如何描述?模糊语义网是如何演化的?这正是本文研究的内容。

1.1.2 研究意义

研究模糊语义网的思想,使得 Web 上的资源具有模糊性,资源间的关系也具有模糊性。并且,这种模糊性的大小会因为外界因素的影响而改变。这使得网络就像人的大脑一样,对事物的认知不全是确定无疑的,而存在很多的不确定性即模糊性,对事物的认知理解会随着种种因素的改变而改变。简单的说,就是网络将能够像人脑一样思考,成为智能网络。

本文选取模糊语义网作为毕业设计研究的内容,在阅读了大量语义网相关资料的基础上,初步提出了模糊语义网的体系结构,并重点研究了模糊语义网体系结构的 RDF(S)层和本体层,研究了具体关系、抽象关系的描述问题,研究了模糊语义网的演化算法。希望通过对模糊语义网的研究,能够为其发展添加新的思想元素,为模糊语义网的研究发展贡献一份力量。

1.2 国内外研究现状

1998 年, Tim Berners-Lee 提出语义网的概念,之后,就成为研究热点。当前关于其的研究,处于起步阶段,我国对语义网的研究较落后。欣慰的是,学者们已经意识到语义网对未来互联网的影响,并开始着手研究语义网相关的技术和应用。

当前,语义网概念仍没形成统一定义,对语义网理解也不同。如语义网是“第三代 Web,其目标是实现机器自动处理信息”^[2];语义网“与现有万维网不一样,它的数据供人类使用,新一代 WWW 中将提供能被计算机处理的数据,使得智能服务成为可能”^[3];语义网的研究目标是“开发能被计算机理解和处理的表达语义信息的语言与技术,从而支持网络环境下广泛有效的自动推理”^[4]。Tim Berners-Lee 对于语义网的定义:“语义网是个网,包含文档或文档的一部分,描述了事物间明显关系,且包含语义信息,利于机器自动处理”^[5]。虽然对于语义网的理解与描述不相同,仍能总结出一些基本特征:语义网不同于 WWW,是它的扩展与延伸;是面向文档所表示的数据;语义网更利于计算机“理解与处理”,并具一定的判断、推理能力。

Berners-Lee 于 2000 年提出语义网的体系结构,目前语义网的体系结构还在不断建设完善中,未形成一个令人满意和信服的理论体系,我国学者也仅做了简要介绍,还没有形成系统阐述。

除了对语义网系统的介绍,我国学者就其关键技术做些有益的尝试与应用研究。在 RDF 技术方面,从 XML/RDF 特点入手,黄伟、任瑞娟等对基于 XML/RDF 的 MARC 和 DC 元数据的描述技术进行了讨论,并给出描述例子^{[6][7]};宓永迪展示了 RDF 的具体应

用^[8]；韩亚洪讨论了 RDF 在关系数据库中是如何存储的^[9]；袁勇智则描述了如何利用 RDF/XML 进行搜索引擎的设计、实现^[10]；杨晓青对 RDF(S)简要介绍的基础之后，讨论如何利用 RDF 建立 Ontology，并给出相应例子^[11]；欧阳晔给出了基于 XML 的 RDF 的查询语言，并在此基础上构建了以 XRQL 为查询语言的 RDF 引擎的体系结构^[12]。

在 Ontology 应用方面，说明如何使用 RDF(S)表示 Ontology 中对象的模型，分析怎样在资源描述框架 RDF(S)中表示本体论中的公理，举了例子说明怎样利用 RDF 技术进行本体的构建；廖明宏介绍基于本体的企业员工能力查询系统，给出了查询算法^[13]；陈文彬从图书馆中的服务出发，从三个层次上分析了本体开发思路^[14]；邓凯在对本体论进行了简要说明，然后提出如何利用本体理论与方法建立知识知识的划分、分类和组织的模型，这方便了知识的存放和智能检索^[15]；楼向英、李景就 Ontology 在电子图书馆方面的应用进行了详细阐述^{[16][17]}；YuWei 在研究本体查询思想的基础上，提出语义网上本体查询和检索的新方法^[18]；Kang Dazhou 提出了支持语义 Web 模糊本体的描述逻辑^[19]。

1.3 本文的主要内容和结构安排

1.3.1 本文的主要内容

本课题研究内容是模糊语义网的体系结构及其推理研究。在深入研究语义网七层体系结构的基础上，初步提出了模糊语义网的体系结构。并重点研究了 RDF(S)层和本体层。本文研究了模糊认知图和模糊本体，并阐述了如何表示具体关系与抽象关系。研究了模糊语义网的演化，演化分为具体关系的演化、抽象关系的演化以及间接关系的演化。本文研究了具体关系和抽象关系的演化。应用反馈神经网络的思想研究具体关系的演化。并讨论了个体隶属度的计算方法。最后做了模糊语义网的演化实验-职称晋升本体演化实验，验证了我们的理论。

本课题的主要工作体现在以下几个方面：

1) 语义网基本原理与技术。学习语义网原理与技术，了解国际及国内对语义网研究的现状及发展的趋势。主要内容有语义网的思想、架构、知识表示模型、表示语言、查询技术和安全等^[20]。对语义网有一个宏观的了解把握，为后续研究奠定基础。

2) 资源描述。当前的 WWW 主要考虑信息的表示是否宜人阅读，而不关注信息能否被机器理解^{[21][22]}。语义网试图扩展，让机器和人共同理解。知识表示语言是语义网的核心，该语言表示的知识具有平台无关性^{[23][24][25]}。RDF(S)层和本体层是语义网核心，用来表示网络信息语义。本文深入探讨研究了 RDF 的概念、数据模型及本体理论和本体语言，将模糊性应用到 RDF(S)层和本体层，使语义网能够描述模糊资源及资源间的模糊关系。

3) 模糊本体。给出了模糊本体的定义、其形式化描述以及如何用模糊本体 Web 语言-FOWL 描述模糊本体。

4) FOWL。即模糊本体 Web 语言，给出了其语法，并定义了属性、个体和类，并用其描述模糊本体，同时研究了个体隶属度计算方法。

5) 本体演化。本体演化是指一个本体面对不断产生的内外界因素变化要能够具备随时间变化的适应能力^[26]。随着时间推移, 由于知识具有内在的变化性, 具有高度的动态性; 领域的变化, 对不同任务的适应性, 以及本体在概念化和关系上的改变均要求给出的知识表示方法要具备很好的动态适应性^{[27][28][29]}。语义网的演化实质上就是本体的演化, 本文研究本体的演化, 深入探讨研究模糊语义网的演化。

1.3.2 本文的章节安排

全文由七章组成。第一章是绪论, 阐述了课题的研究背景, 针对海量信息处理存在的问题, 提出了该课题的研究意义与内容; 第二章是模糊语义网的体系结构, 在研究了语义网的原理及体系结构后, 又深入学习了模糊语义网的思想, 初步给出了模糊语义网的体系结构, 并阐述了各个层的功能及作用, 叙述了模糊语义网的研究问题。第三章为具体关系及其描述, 学习了模糊认知图、资源描述框架 RDF(S), 模糊资源描述框架 FRDF 以及它们的互表示性问题。第四章为抽象关系及其描述, 研究了本体、模糊本体、模糊本体语言 FOWL 的语法以及表示问题; 第五章为模糊语义网的演化研究, 模糊语义网的演化包括了具体关系的演化、抽象关系的演化以及间接关系的演化。本章在研究神经网络的基础上, 重点阐述和研究了具体关系及抽象关系的演化。第六章为职称晋升本体演化实验, 通过实验验证提出的理论及对实验结果的分析与总结。第七章为结论, 是对论文工作所作的总结, 其中指出并分析了尚未解决或者不完善的部分, 并对将来的工作进行了展望。

2 模糊语义网的体系结构

2.1 语义网概述

“语义”是文本的含义。语义网好比一个巨型的大脑，由数据库智能化程度很高，协调能力非常强大的各部分组成。在语义网上连接的每台电脑，都能分享人类所有科学和艺术等知识。它不仅能够理解词语和概念，还能够理解和解释它们之间的逻辑关系。

在语义网中，网络不但能够连接各个文件，还能够识别文件里所传递的信息。

2.1.1 语义网的概念

语义网的研究处于起始阶段，对语义网的概念仍没形成统一的定义，对它的理解表述也各有不同。W3C 是这样阐述的：将机器能理解的数据发布在 Web 上，只有让 Web 成自动工具和人能够共享和处理的数据平台，其潜能才能全部发挥^{[30][31]}。语义网创始人 Tim Berners-Lee 对语义网的定义是这样的：语义网是个网，包含了文档和文档的一部分，描述了事物间明显关系，且含语义信息，以便于机器自动处理^{[32][33]}。

2.1.2 语义网的特征

尽管对语义网的理解与描述不同，但仍能看出语义网的一些基本特征：

- 1) 语义网不同于现有 WWW，是 WWW 的扩展与延伸；
- 2) WWW 是面向文档而语义网是面向文档所表示的数据；
- 3) 语义网更加利于计算机“理解和处理”，并具一定判断、推理能力。
- 4) 语义网目标是让信息能被机器理解，从而完成信息的自动处理，以适应 Web 资源的快速增长，更好的为人类服务。
- 5) 在语义网环境下，Web 上的数据不仅能显示，而且可被机器自动处理、集成和重用。

2.2 语义网的架构

根据 Berners-Lee 设想，语义网体系结构是分层的，如图 2.1 所示。是一个功能不断增强的层次化结构，由七个层次构成。

1) URI + Unicode (统一资源标识+国际码)

该层是表示语义网对象和国际字符集的基本手段。URI 是 Web 的核心，能唯一地标识 Web 上任一资源，思想是在需要的时候连接引用，不需要对资源进行拷贝和集中管理。

2) XML + NS + XML Schema (可扩展标记语言+名字空间+可扩展标记语言大纲)

此层定义了语法互操作标准。XML 提供了文档结构化的语法，根据不同目的同一个文档可以有多种不同表现形式。XML 命名空间是名称的集合，用于文档元素和属性名的

有效验证, 由 URI 来标识。XML 是底层数据交换格式, 它仅解决了文档内容的次序结构问题, 并未解决文档内容的语义、联系问题。标签的具体含义的定义和互操作要交给上一层解决。

3) RDF + RDF Schema (资源描述框架+资源描述框架大纲)

本层用来定义和描述语义网上的资源。XML 虽然实现了文档结构化, 但不包含语义信息。RDF 提供简单语义, 其属性可看作是资源的属性, 同时也表达资源间的关系。

RDF Schema 为 RDF 模型提供了一个基本的类型系统, 目的是定义资源属性、被描述资源的类, 并提供检测机制。

4) Ontology vocabulary (本体词汇汇集)

该层定义不同概念间的关系, 以支持词汇演化。本体层 (Ontology) 提供一个明确的形式化语言, 以准确定义术语语义及术语间的关系。

5) Logic、Proof & Trust (逻辑、证明和信任)

本层提供公理描述框架, 公理和逻辑推理规则, 使智能服务成为可能。Proof 层执行规则并做出评估, 为智能代理间相互验证交换数字签名提供支持。Trust 层为应用程序是否信任一个给定的证明提供检测机制。

6) Digital Signature (数字签名)

就是一段数据加密块, 可用它来验证某个信息是否由可信任的来源提供。它是实现 Web 信任的关键技术。

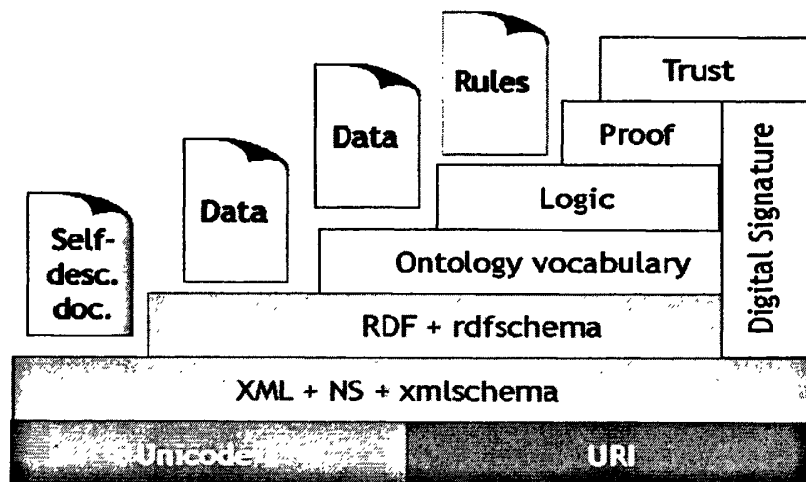


图 2.1 语义网体系结构

2.3 语义网的研究问题

目前, 语义网研究问题主要有:

1) 语义网知识表示模型。采用本体作为表示知识的模型。本体形式化的定义了领域内共同认可的知识。因此, 创建和管理本体是实现语义网上知识表示的基础。

2) 语义网逻辑基础。描述逻辑是语义网的逻辑基础。语义网语言的表达能力离不开

相应的描述逻辑，推理问题也可由描述逻辑推理算法实现。

3) 语义网表示语言。提供了领域信息的描述基础，是语义网的核心。

4) 语义网查询技术。是语义网最基本的技术，语义网的各种应用都离不开查询技术的支持。

5) 语义网的安全。安全是语义网实现的必要条件。安全包含每一层次的安全以及互用性的安全。

2.4 模糊语义网的提出

语义网的构想是在 Web 中引入语义知识表示。因此，怎样标识语义信息对于语义网就显得尤其关键。RDF(S)在 XML 的基础上提供有限的语义描述能力，但还需要添加用于描述属性和类型的词汇。作为本体描述语言的 RDF/RDFS 是本体层研究的基础，RDF 主要用于对信息资源的标识和描述，RDFS 则主要用于本体的定义和描述。

本文从本体描述语言 RDF/RDFS 入手，在对其深入研究的基础上，认为 RDF/RDFS 在对 Web 资源的描述上可以进行一定的改进，以增强其描述 Web 资源的能力。因此，这就是模糊语义网的最初想法，就是在 RDF/RDFS 的基础上，对 RDF 数据模型结构作一些改进，然后对语义网层次结构中 RDF(S)及以后的各层也进行相应的修改，重点研究了 RDF(S)层和本体层，引入相应的模糊权值，提出模糊本体的概念，给出了模糊本体 Web 语言-FOWL。这样就达到了模糊语义网的预期目标——使 Web 上的数据能以一种可以被机器所理解的方式定义和联系起来，并且使得数据本身具有模糊性，数据间的关联程度具有模糊性，它不仅以显示为目的，在 Web 上信息是发展变化的，信息的语义及相互间的联系也将发生着变化。我们知道信息是互相表示的（即一个概念离开其它相关概念是没有意义的，称为信息的互表性），且信息是不断变化的（即概念本身或其属性受到外界或自身内在的变化而使得概念或概念间的关系发生了变化，称为信息的演化性），导致了信息的不确定性。要比较真实的表示现实世界，就要把信息的互表性和演化性表示出来，因此我们在语义网的体系结构中引入了表示信息的互表性和演化性的成份，提出了模糊语义网的概念。也就是说，未来的语义网，就像人一样，具有智能性，语义网上的资源就好像是人脑中的知识，随着年龄的增长，社会环境的变化，脑中的知识会有一个不断改变、演化的过程。

2.5 模糊语义网的体系结构

在研究了语义网体系结构的基础上，模糊语义网的体系结构是在其第三层资源描述框架 RDF(S)层引入模糊权值，从而影响到了本体层、逻辑层，使得以上各层都发生了相应的改变。

模糊语义网的体系结构可用图 2.2 表示，仍为七层，第一层是：URI + Unicode（统一资源标识+国际码）；第二层是：XML + NS + XML Schema（可扩展标记语言+名字空间+

可扩展标记语言大纲); 第三层是: FRDF(S) (资源描述框架+资源描述框架大纲); 第四层是: Fuzzy Ontology(模糊本体层); 第五层是: Fuzzy Logic (模糊逻辑层); 证明层 Proof 和信任层 Trust 分别为第六层和第七层。

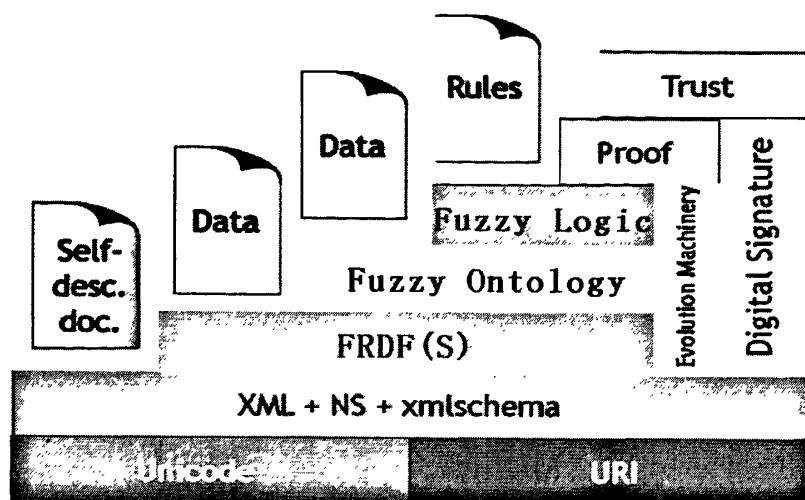


图 2.2 模糊语义网体系结构

1) URI + Unicode (统一资源标识+国际码)

该层是表示语义网对象和国际字符集的基本手段。URI 是 Web 的核心，能唯一地标识 Web 上任一资源，思想是在需要的时候连接引用，不需要对资源进行拷贝和集中管理。

2) XML + NS + XML Schema (可扩展标记语言+名字空间+可扩展标记语言大纲)

此层定义了语法互操作标准。XML 提供了文档结构化的语法，根据不同目的同一个文档可以有多种不同表现形式。XML 命名空间是名称的集合，用于文档元素和属性名的有效验证，由 URI 来标识。XML 是底层数据交换格式，它仅解决了文档内容的次序结构问题，并未解决文档内容的语义、联系问题。标签的具体含义的定义和互操作要交给上一层解决。

3) 具体关系描述层 FRDF (S) (模糊资源描述框架)

定义 1: 具体关系: 事物间本有的、一目了然的关系。如对于一个个体人，他有属性名字、性别、年龄。

本层用来描述和定义语义网上的模糊资源。XML 实现了文档结构化，但文档信息并不包含语义。FRDF 数据模型提供简单语义，FRDF 属性可以看作是资源的属性，同时又表达了资源间关系亲密程度。

4) 抽象关系描述层 Fuzzy Ontology vocabulary

定义 2: 抽象关系: 指人们为了研究方便而主观定义的事物间的关系。如个体、属性、部分、类、属于、等价、交、并、补等。

抽象关系层用来定义不同概念间关系，支持词汇的演化。虽然 FRDF(S)能够定义对象的属性和类，并且还提供类的泛化等简单语义，但他不能够明确表达描述属性或类的术语

的含义及其间关系。抽象关系描述层提供一个明确的形式化语言,以准确定义术语语义及其间的关系以及关系的程度。

5) 间接关系描述层 Fuzzy Logic

定义 3: 间接关系: 概念间接关系是指在具体关系描述层和抽象关系描述层中没有直接给出的关系,就要通过推理给出,这就需要提供公理和逻辑推理规则,当然也是模糊的。

这就需要提供公理和逻辑推理规则,当然也是模糊的。

6) 演化机制 Evolution Machinery

我们知道,在大自然界中,事物和事物间的关系都是不断变化的。比如,对于每一个个体人,从出生到死亡,发生了一系列复杂的变化,这只是大自然中演化(自然演化)的非常渺小的一个例子。那么在计算机中,我们如何模拟这种变化呢?遵循什么规律呢?我们就要提及到演化机制和演化算法。

模糊语义网体系结构中的第三层具体关系描述层 FRDF(S)、第四层抽象关系描述层 Fuzzy Ontology vocabulary、第五层间接关系描述层 Fuzzy Logic 是我们的语义描述层,演化机制贯穿这三层。

7) Proof & Trust(证明和信任)

Proof 层执行规则并做出相应的评估,为智能代理间互相验证交换数据的签名提供支持。Trust 层为应用程序是否信任一个给定的证明提供检测机制。

8) Digital Signature(数字签名)

是一段数据加密块,可用它来验证某个信息是否由可信任的来源提供。它是实现 Web 信任的关键技术。

2.6 模糊语义网的研究问题

前面我们讨论了语义网的研究问题,那么除了这些问题,我们模糊语义网还需要重点研究或进一步完善的问题都有哪些呢?

1) 模糊语义网及其演化研究

模糊语义网的演化包括具体关系的演化、抽象关系的演化以及间接关系的演化,目前我们只研究了具体关系和抽象关系的演化,而对于间接关系的演化还没有做,有待研究。

2) 模糊语义网的体系结构及其推理研究

虽然前面我们已经叙述了模糊语义网的体系结构,可以说只是个概括,需要进一步的细化,进一步的完善。比如逻辑、证明都要贯穿在语义层,数字签名、安全的都需要考虑。

3) 模糊本体生成工具(就象 PROTEGE)的研究与设计

对现有的本体生成工具(如 PROTEGE)进行扩展,能够建立模糊本体。

4) 模糊语义网生成工具(就象 JENA)的研究与设计

对现有的语义网生成工具(如 JENA)进行模糊扩展,能够生成模糊语义网。

5) 模糊语义网解释系统的研究与设计

模糊语义网是为了机器可理解而建立的错综复杂的网状系统,怎样才能有利于人类理解和应用呢?这就需要对模糊语义网解释系统进行研究。

6) 模糊语义网的应用研究(通用本体、领域本体和个人本体,如搜索引擎、单位本体和课程本体等)

比如基于模糊语义网的搜索,类似现在的百度,不同的是我们搜索到的结果的每个词条都与关键字之间有亲密程度、由大到小排列,同时还考虑到不同个体的偏好。还有我们的课程本体、职称晋升本体等在教学中都有应用前景。

3 具体关系及其描述

定义 1 中指出,具体关系是指事物间本有的关系或者说是一目了然的关系。比如对于一个个体人,他有属性名字、性别、年龄,这些都是很显然明了的关系,我们把这些关系称为具体关系。模糊语义网中的具体关系是指具体的模糊关系,也就是具体关系的明晰化,可以表述关系的亲密程度、比重等。为此我们给出了模糊资源描述框架 FRDF。

3.1 模糊认知图

我们的模糊资源描述框架 FRDF(S)是在研究了模糊认知图原理之后,将其原理应用到资源描述框架 RDF(S)中,为资源描述框架三元组中的属性添加了权值,用以表示模糊性,形成了模糊资源描述框架 FRDF(Fuzzy Resource Description Framework),以提高复杂知识及语义的表示与处理能力。下面对模糊认知图做简要介绍:

Axelord 于 1976 年提出认知图,概念与概念间的关系是其组成部分。概念反映系统的属性、性能与品质、状态。概念间关系表示的是概念间的关联、影响关系,其关联强度用数值表示,即弧的权值^[34]。Axelord 的认知图作为一种新的定性推理和知识表示的技术方法应用在表达和推理系统。作为重要的智能信息处理工具,它能表示出具有反馈的动态因果系统,提供了用于支持先验知识的自适应行为的一个有效的软计算工具。

Kosko 在此基础上提出的模糊认知图是结合了神经网络与模糊逻辑技术,其推理和知识表示能力强。FCM 理论在医学、军事、政治、社会关系和信息系统、网络管理、多 Agent 系统、等领域中得到了广泛应用^{[35][36]}。

FCM 将知识存储于概念间关系来表示模糊推理,通过网络中概念节点间相互作用来模拟系统的行为。FCM 初始状态空间由初始条件决定,通过顶点函数传播,直至达到静态模式。当 FCM 达到了稳定循环状态或固定点,就完成因果关系推理。

模糊认知图定义如下:

定义 4 FCM 的拓扑结构是一个三元序组 $U=(V,E,W)$, 其中 $V=\{v_1,v_2,\dots,v_n\}$ 表示 FCM 概念节点的集合, $E=\{\langle v_i,v_j \rangle | v_i,v_j \in V\}$ 所有节点间的因果关联有向弧 $\langle v_i,v_j \rangle$ 表示节点 v_i 对 v_j 有因果关联, $W=\{W_{ij} | W_{ij} \text{ 是有向弧 } \langle v_i,v_j \rangle \text{ 的权值}\}$ (即 W_{ij} 表示结点 v_i 对 v_j 的关联强度)。每个节点都有状态空间, $V_{ci}(t)$ 表示节点 v_i 在 t 时刻的状态值。如果 $w_{ij} > 0$, 则 v_i 对 v_j 有正影响; 如果 $w_{ij} < 0$, 则 v_i 对 v_j 有负影响; 如果 $w_{ij} = 0$, 则表明 v_i 对 v_j 无影响, 此时 v_i 到 v_j 不联边。FCM 在时间 t 的状态可以方便地用向量函数表示:

$$\Phi_u(t) = (v_1(t), v_2(t), \dots, v_n(t)) \quad (3.1)$$

FCM 的边权矩阵如下:

$$Wu = \begin{pmatrix} \dots & \dots & \dots \\ \dots & W_{ij} & \dots \\ \dots & \dots & \dots \end{pmatrix} \quad (3.2)$$

其中, W_{ij} 表示结点 v_i 对 v_j 的关联强度。

FCM 推理模型为:

$$V_{C_j}(t+1) = f \left(\sum_{\substack{i \neq j \\ i \in E}} V_{C_i}(t) w_{ij} \right) \quad (3.3)$$

式中 $V_{C_i}(t)$ 为概念结点 C_i 在 t 时刻的值, $V_{C_j}(t+1)$ 为结果概念结点 C_j 在 $t+1$ 时刻的状态值, E 为与 C_j 有邻接关系的概念结点集合, f 为阈值函数, 可以是二值的、S 型的、模糊集合或概率函数。

进一步, 由矩阵乘法得到, FCM 在时间 $t+1$ 收到的输入为:

$$(V'_1(t+1), \dots, V'_n(t+1)) = \Phi_U(t) \times W_U \quad (3.4)$$

从而, FCM 在时间 $t+1$ 得到输出:

$$\Phi_U(t+1) = (V_1(t+1), \dots, V_n(t+1)) = (f(V'_1(t+1)), \dots, f(V'_n(t+1))) \quad (3.5)$$

FCM 是有符号、带反馈的有向网络, 其结构如图 3.1 所示^[24]:

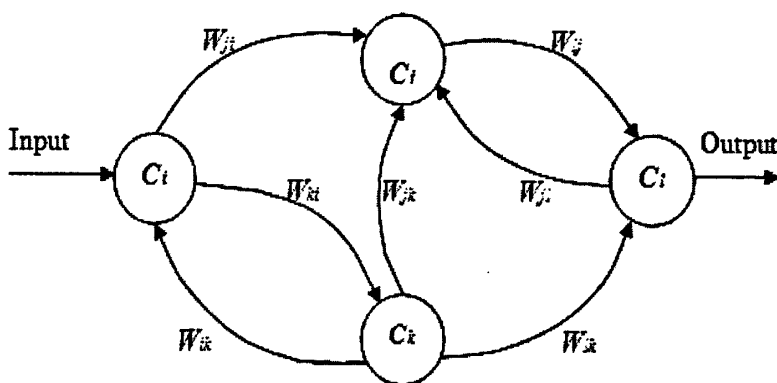


图 3.1 FCM 结构示意图

从图中可以看出, FCM 由网络节点间的连接弧组成。其中, 每个网络节点都有其状态值, 刻画了 FCM 环境中物元属性; 节点间有向连接弧代表了物元间的因果关系。物元间的关系权值取为 $[-1, 1]$ 的等间隔数值, 能对不同类型和精度的模糊信息进行处理和描述。^{[25][26]}

3.2 资源描述框架 RDF

目前的 Web 主要支持对文本内容的浏览和搜索^[37]。随着 Web 信息的增加,该模型无法适应大量信息的交换和处理工作。语义网需要新模型以支持对 Web 信息源和服务以及职能应用的统一访问,并用标准机制去交换数据和处理不同的数据语义。

本体描述语言 RDF 已经成为 W3C 的建议。RDF 是元数据处理和操作的基础。它提供了应用程序协同工作能力。语义 Web 处理的是具有固定格式的信息。RDF 是描述 Web 资源的元数据,属于语义 Web 中的元数据层,是关于任何网络资源的元数据框架。

3.2.1 RDF 的含义

资源描述框架提供了通用的数据模型支持对 Web 资源描述。其含义如下:

- 1) 资源。RDF 可处理的 Web 资源含义很广,包括在 Web 上一切被命名、具有 URI 引用的东西。
- 2) 描述。对资源特性的声明,包括资源本身属性及资源间的关系。
- 3) 框架。与被描述资源和领域无关的通用模型。RDF 定义一种机制用来描述非特定领域资源。

3.2.2 RDF 模型

RDF 的设计思想:被描述资源具有一些特性,有值;既可是文字也可以是其他资源;如果是资源,特性就可以看成是资源间关系。该 RDF 数据模型由四种基本对象类型组成:

- 1) 资源。Web 上以 URI 表示的所有事物均为资源。
- 2) 文字。字符串或数据类型的值。RDF 无自己的数据类型定义机制,允许使用单独定义的数据类型。
- 3) 特性。特性用来描述资源的属性或关系。可定义它的许可值、描述的资源类型及特性间的关系。
- 4) 声明。一个特定资源加上特性、特性值就成为一个 RDF 声明。

RDF 声明用一个三元组表示,即(资源,特性,特性值),也就是(主体,谓词,客体)。

RDF 模型结构是声明三元组集合,可用具有节点和边的 RDF 图表示。RDF 图节点是主体与客体;边由谓词担当,边的方向由主体指向客体。

例如,“Tom 电子邮件为 Tom@email.org”这条资源用一个 RDF 三元组表示为
ex:person1023 ex:email “Tom@email.org”

这个三元组可用 RDF 图表示,如图 3.2 所示。

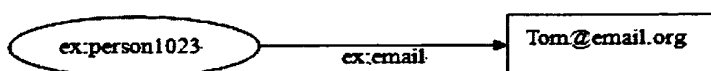


图 3.2 三元组的 RDF 图

RDF 图可表示多个三元组。例如：有 main.html 和 index.html 两个网页，创建它们的是 name 为 Tom、email 为 Tom@email.org 的人。这句话可以用四个 RDF 三元组表示：

```
ex:main.html dc:creator ex:person1023
ex:index.html dc:creator ex:person1023
ex:person1023 ex:email "Tom@email.org"
ex:person1023 ex:name "Tom"
```

此声明用 RDF 图（见图 3.3）表示。

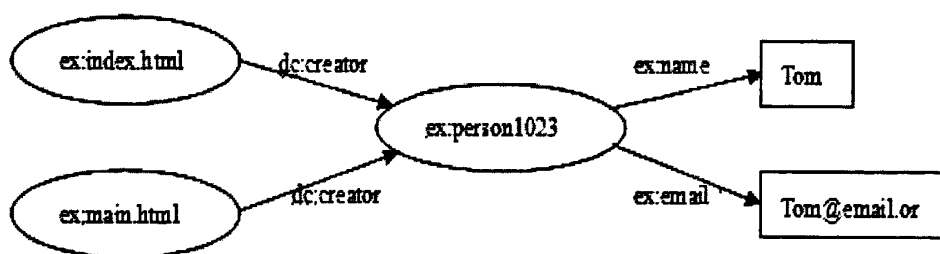


图 3.3 包含多个三元组的 RDF 图

RDF 图虽直观，但不便机器处理；三元组表示简略清楚，但不便网络上交换；RDF/XML 语法是用基于 XML 的语法来编码 RDF 的，W3C 推荐的描述 RDF 语法。

3.2.3 RDF/XML 语法

RDF 提供 RDF/XML 的 XML 语法来书写和交换 RDF 图模型。给出一张 RDF 图，RDF/XML 这样书写：

- 1) 所有空节点均指定空节点标示符。
- 2) 对每个三元组，取其主体对应节点，生成一个 `rdf:Description` 元素。

若该节点有 URI 引用，则 `rdf:Description` 元素使用 `rdf:about` 属性，值为该 URI 引用，若为空节点，则 `rdf:Description` 元素用 `rdf:nodeID` 属性，值为空节点标示符。

- 3) 创建谓词 URI 引用的子元素。若客体是文字，则该子元素内容为客体，或者是一个以客体 URI 引用为值的 `rdf:resource` 属性，或者是有一个以客体的空节点标示符为值的 `rdf:nodeID` 属性。

上述方法生成的 XML 语句冗长，不适合处理。为此 RDF/XML 提供了简写语法。

同个主体的多个声明可并为一个 `rdf:Description` 元素，多个谓词将转化为该元素的子元素，客体就成为对应谓词元素的属性或者是内容。

若一个声明的主体是另一个的客体，则前者可嵌套至后者的客体位置。

若空节点 `rdf:Description` 元素被嵌套，再其他地方不会用到该空节点，则对应的 `rdf:Description` 元素标记可以删除。

下面举例说明, 用 RDF/XML 语法表达图 3.3 的 RDF 图:

```
<? xml version="1.0"?>
<rdf:RDF xmlns:rdf=http://www.w3.org/1999/02/22-rdf-syntax-ns#
    xmlns:dc="http://purl.org/dc/elements/1.1/"
    xmlns:ex="http://www.example.org/">
  <rdf:Description rdf:about="http://www.example.org/index.html">
    <dc:creator rdf:resource="http://example.org/people1023"
      ex:name="Tom" ex:email="Tom@email.org"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://www.example.org/main.html">
    <dc:creator rdf:resource="http://example.org/people1023"
      ex:name="Tom" ex:email="Tom@email.org"/>
  </rdf:Description>
</rdf:RDF>
```

3.3 模糊资源描述框架 FRDF

资源描述框架-RDF 数据模型的核心是用(主体、谓词、客体)三元组来定义 Web 上的资源, 如果三元组中的客体是资源的话, 这个三元组就描述了俩资源间的关系。此时, 所描述的关系是静态的。不能够表示出来关系亲密程度和关系程度随时间、空间的改变而改变的情况, 认为关系是确定不变的。

3.3.1 模糊资源描述框架 FRDF

通过对模糊认知图的研究学习, 将模糊认知图原理应用到 RDF 中。在(主体、谓词、客体)三元组中的谓词部分加入一个表示权值的元素, 表示谓词的模糊性, 即主体与客体间关系的模糊度。在研究的基础上提出一个新的概念——模糊资源描述框架(FRDF, Fuzzy Resource Description Framework), FRDF 模型用三元组(主体、(谓词、权值)、客体)来定义 Web 上的模糊资源和描述资源之间的模糊关系。

FRDF 数据模型由五种基本对象类型组成:

- 1) 资源。Web 上以 URI 表示的所有事物均可称为资源。
 - 2) 文字。字符串或数据类型的值。
 - 3) 特性。特性用来描述资源属性或关系。可定义它的许可值、描述的资源类型及特性间关系。
 - 4) 权值。用来描述资源具有某种特性、属性及资源间关系的可信度或亲密程度(即模糊性)。
 - 5) 声明。一个特定资源加上特性、特性值就成为一个 RDF 声明。
- Web 上有很多资源包含相同的信息或知识, 但是相同的信息或知识在每个资源中所

占比重有所不同。用户在查询这些信息时,系统无法比较哪个资源包含的信息既详细又准确。只能够将含有这些信息的资源全部返回到客户端,查询效率不高。利用 FRDF 来描述 Web 上的资源,就可以改善这个问题。可以用 FRDF 中的权值来表示相同信息或知识在各资源中所占的比重。这样的话,当用户查询信息时,系统会根据权值来比较资源包含这些信息的多少及准确度等。这样就提高了查询的效率。

3.3.2 利用模糊认知图表示 FRDF

在提出模糊资源描述框架 FRDF 的概念之后,本文采用模糊认知图 FCM 来表示它。基于模糊认知图 FCM 的局限性,本文将模糊认知图 FCM 做了一些改进以能够支持 RDF 实现更强大的功能。

1) 定义 4 中 $E = \{\langle v_i, v_j \rangle | v_i, v_j \in V\}$ 为全部节点间的因果关联有向弧,有向弧 $\langle v_i, v_j \rangle$ 表示节点 v_i 对 v_j 有因果关联。指定节点间关系为因果关系,在资源描述框架应用中具有局限。所以令 E 为与 W 相对应的矩阵,其中元素 $\langle v_i, v_j \rangle$ 表示节点 v_i 与 v_j 间的关系,取值为关系名。资源描述框架主要描述资源间关系,可应用 FCM 来表示 FRDF 中主客体间的简单模糊关系。

2) 在模糊认知图的定义中,仅定义了两节点间关系的一维性。在实际应用中,两个事物间的关系可能不止一种,往往存在多维关系。为此,给出多重模糊认知图定义:

定义 5 若将定义 4 中的 W 改成广义的模糊矩阵

$$W = \begin{pmatrix} W_{11} & W_{12} & \cdots & W_{13} \\ W_{21} & W_{22} & \cdots & W_{23} \\ \cdots & \cdots & & \cdots \\ W_{31} & W_{32} & \cdots & W_{33} \end{pmatrix} \quad (3.6)$$

其中 $W_{ij} (i=1,2,\dots,n; j=1,2,\dots,n)$ 是元素个数不超过 m 的隶属度集合

$$W_{ij} = \{W_{ij1}, W_{ij2}, \dots, W_{ijm}\} \quad 0 \leq W_{ijk} \leq 1, \quad (k=1,2,\dots,m)$$

则称 $U = \{V, E, W\}$ 为一个论域 V 上的多重模糊认知图。

改进的模糊认知图可与 FRDF 相互表示和转化,FCM 定义中,节点集合 V 表示了 FRDF 中的资源;节点间的有向弧 E 表示谓词;有向弧的权值 W 即对应 FRDF 的权值。

例如,有 index.html 和 main.html 两个网页,创建者均是 email 为 Tom@email.org、name 为 Tom 的人。可表示为四个 FRDF 三元组:

ex: main.html dc: creator (0.5) ex: person123

ex: index.html dc: creator (0.8) ex: person123

ex: person123 ex: email(0.7) Tom@email.org

ex: person123 ex: name(0.9) "Tom"

其中 `dc: creator` 是都柏林核心词汇汇集定义的作者特性, 权值 0.8、0.9、0.5、0.7 表示关系的模糊性。可用改进后的模糊认知图表示, 如图 3.4 所示。

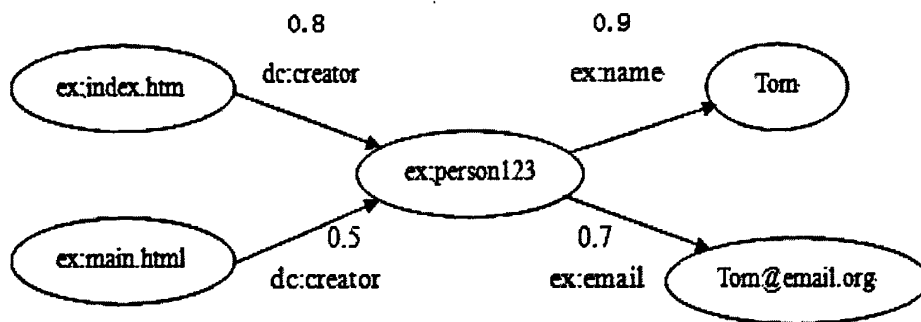


图 3.4 模糊认知图对 FRDF 的表示

4 抽象关系及其描述

定义 2 中指出,抽象关系是指人们为了研究方便而主观定义的事物间的关系。如个体、属性、部分、类、属于、等价、交、并、补等。这些都是为了研究方便而定义的关系,我们把这些关系称为抽象关系。模糊语义网中的抽象关系是指抽象的模糊关系,也就是抽象关系的明晰化,可以表述关系的亲密程度、比重等。为此我们给出了模糊本体 WEB 语言 FOWL 的语法,并讨论了如何用 FOWL 定义属性、个体及类。

4.1 本体

语义网构想在 Web 引入语义知识表示,保证 Web 页面能被机器理解并且自动处理。因此,如何标识语义信息就显得非常重要。RDF (S) 在 XML 的基础上提供很有限的语义描述能力。语义信息模型本体是描述语义知识的建模手段,它形式化的定义领域内共享的知识,是语义网的核心。

4.1.1 本体的定义

本体在 20 世纪 60 年代就在计算机领域普遍使用。第一个令人信服的本体定义是 1993 年 Gruber 给出的,即“本体是概念模型的明确的规范说明”。后来, Borst 在此基础上,给出了另一个定义:“本体是共享概念模型的形式化说明”。Studer 等人对上述两个定义深入研究后,认为本体是“共享概念模型的明确的形式化规范说明”,包含了四层含义:

概念模型:过抽象客观世界中的概念而得到的模型,独立于具体环境状态。

明确:念和概念的约束都有无歧义的定义。

形式化:通过本体语言编码,使本体能让计算机可读,并可以被处理。

共享:同认可的知识,反映相关领域中公认的概念集。

以上是本体定义的文字描述,下面给出本体的形式化定义。

本体定义为一个七元组 $O = (C, A^C, R, A^R, H, I, X)$, 其中 C 是概念集合; A^C 是概念属性集合; R 是关系集合; A^R 是关系属性集合; H 是层次的集合; I 是实例集合; X 是公理集合。

1) 概念。亦称为类,从语义上讲,是对现实世界个体的抽象,其定义一般包括概念名称及对该概念的自然语言描述。

2) 概念的属性。如果 c 是 C 中的概念,它的属性可表示为 $A^C(c)$ 。概念间之所以有差异是由于属性的不同,才对应不同的个体集合。

3) 关系。一个关系通常包含定义域、值域两部分,限定了关系适用的范围。

4) 关系的属性。描述对关系的进一步限制。

5) 层次。可定义在概念、属性和关系之上。。

6) 实例。现实世界中具体、唯一的个体，对应本体中一个或多个概念，具有概念描述的属性和属性值。

7) 公理。代表领域知识中的永真断言，例如，声明关系“Teach”和“Taught-by”是互逆的。

4.1.2 本体语言 OWL

通过 RDF(S)可表示些简单本体，描述某些本体知识。但用作一般本体表示语言时表达能力显得较弱，还缺少一些特性，因此需要描述能力更为富的本体语言，同时要权衡表达能力和有效的推理支持。为此，W3C 提出 OWL 扩展 RDF(S)。

OWL 语言是定义和示例 Web 本体的语言，是 RDF 扩展得到的，既是 Web 标识语言，又是本体描述语言，可以在 Web 上发布和共享本体。W3C 将 OWL 定义为三种不同的子语言（OWL Lite、OWL DL、OWL Full）。

OWL 语言层次如图 4.1 所示：

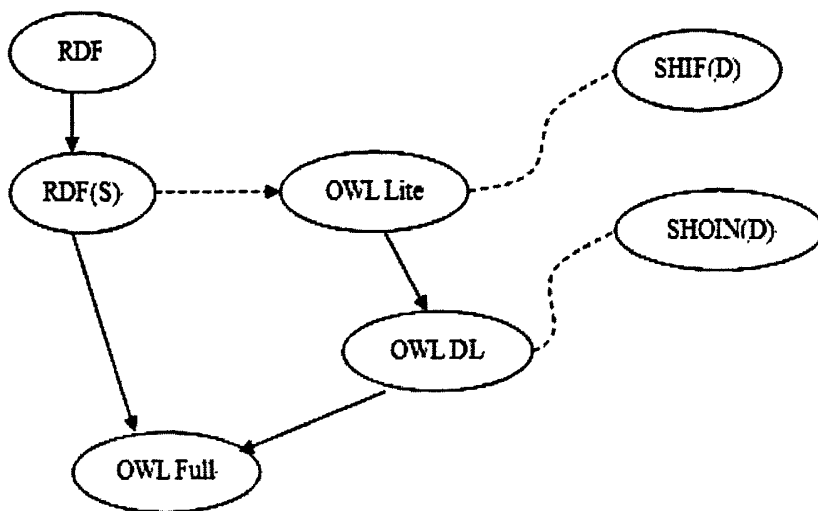


图 4.1 OWL 语言层次

图中 OWL Lite 和 OWL DL 对应描述逻辑 SHIF(D)与 SHOIN(D)，在数据类型的处理上有限制。OWL Lite 是对受限 RDF(S)版本的扩展，OWL DL 是对 OWL Lite 的扩展。OWL Full 是对 RDF(S)的扩展，且包含 OWL DL 所有成分。那么，所有的 RDF 文档都是 OWL Full 文档，但仅有一些 RDF 文档是合法的 OWL Lite 和 OWL DL 文档。

4.1.3 OWL 语法

OWL 建立在 RDF(S)基础之上，利用了 RDF/XML 语法。OWL 本体是 RDF 文档，包含在 `rdf:RDF` 元素中。OWL 本体以关于本体的声明作为开始，包含注释和导入其他本体

等内容,是本体头部。`owl:Ontology` 是用来声明关于当前文档的 OWL 元数据,是本体头部开始。本体头部以`</owl:Ontology>`结束,后面是构成本体的定义,最终由`</rdf:RDF>`终止定义。

下面从简单类、复杂类的定义、属性的特性和约束以及本体映射方面介绍 OWL 语法。

1) 简单类、个体和属性

a、类。OWL 类的定义由两部分组成:引用一个名称及一个限制列表。如下面的定义:

```
<owl:Class rdf:ID="Person"/>
<owl:Class rdf:ID="Male"/>
<owl:Class rdf:ID="Man">
<rdfs:subClassOf rdf:resource="#Person"/>
</owl:Class>
<owl:Class rdf:about="#Man">
<rdfs:subClassOf rdf:resource="#Male"/>
</owl:Class>
```

这里定义了 `Person`、`Male` 和 `Man` 三个类,并且声明 `Man` 是 `Male` 和 `Person` 的子类。

假设该本体的名称空间为 `ex`,对应的抽象语法可以表示为:

```
Class (ex:Person partial owl:Thing)
Class (ex:Male partial owl:Thing)
Class (ex:Man partial owl:Thing)
subClassOf (ex:Man ex:Person)
subClassOf (ex:Man ex:Male)
```

`Class (A partial B)`表示定义一个类 `A`,且 `A` 是 `B` 的子类。OWL 中的所有个体都是类 `owl:Thing` 的成员,因此用户自定义的类都是 `owl:Thing` 的一个的子类,在 RDF/XML 语法中不用写出。OWL 也可以定义空类,用 `owl:Nothing` 表示。

定义一个类的抽象语法也可以写成:

```
Class (ex:Man partial ex:Person ex:Male)
```

上式中将 `Man` 的超类列在 `partial` 之后,说明 `Man` 是所有这些类的子类。

b、个体。本体中引入个体,需声明它是某个类的成员,可用 `rdf:type` 为个体声明其所属的类。例如:

```
<owl: Thing rdf: ID="Tom"/>
  <rdf: type rdf: resource="#Person"/>
  <rdf: type rdf: resource="#Male"/>
</owl:Thing>
```

此语法声明了个体 `Tom`,他是 `Person` 和 `Male` 的实例。用抽象语法可以表示为:

Individual (ex: Tom type (ex: Person) type (ex: Male))

c、属性。属性是二元关系，表示类的实例间的关系或类实例与文字间的关系。

对象属性用 owl:ObjectProperty 定义，并可以用 rdfs:domain 和 rdfs:range 指明它的定义域和作用域，还可以将一个属性定义为某个已有属性的子属性。例如：

```
<owl: ObjectProperty rdf: ID="createBook">
    <rdfs: subPropertyOf rdf: resource="#create"/>
    <rdfs: range rdf: resource="#Book"/>
</owl: ObjectProperty>
```

用抽象语法表示如下：

```
ObjectProperty (ex: createBook super (ex: create)
                domain (ex:Person) range(ex:Book) )
```

其中 super、domain 和 range 均可出现多次，指定多个父属性、定义域和值域，含义是它们的合取。

属性的使用如下：

```
<owl: Thing rdf: ID="DiaryOfTom"/>
    <ex: Man rdf: ID="Tom"/>
    <ex: createBook rdf: resource="#DiaryOfTom"/>
</ex: Man>
```

这时可以推断出 DiaryOfTom 是一本书，即 Book 类的实例。用抽象语法中的 value 可以表示为：

```
Individual (ex: Tom type (ex: Man) value (ex: createBook ex: DiaryOfTom))
```

value 可出现多次，说明个体的多个属性。

2) 复杂类

OWL 提供类构造子用于创建复杂类，这些构造子包括集合的并、交、补等操作。分别被命名为 owl:unionOf、owl:intersectionOf 和 owl:complementOf。另外，OWL 可通过使用 owl:one of 构造子枚举类的所有实例来定义一个类；通过 owl:disjoinWith 声明多个类是不相交的；还可以通过 owl:Restriction 定义匿名类。

3) 属性的特性和约束

a、属性特性

OWL 属性可声明具有函数性、传递性、对称性等特性。通过声明属性属于特定属性类来完成这样的声明。用 $P(x,y)$ 表示 x 的 P 属性值为 y ，可以理解为 x 和 y 间存在 P 关系。

b、属性约束

前面属性约束主要是对属性全局定义域和值域的约束，很多时候属性的值域是根据上下文变化的。

(1) owl:allValuesFrom 属性约束。要求每个有指定属性的类实例，该属性的值必须由 owl:allValuesFrom 从句指定的类的实例。

(2) **owl:someValuesFrom** 属性约束。要求实例至少有一个指定属性的值是指定的类的实例。

(3) 属性的基数约束。如 ≥ 2 hasParent 和 ≤ 2 hasParent 用抽象语法表示为

restriction (ex:hasParent minCardinality(2))

restriction (ex:hasParent maxCardinality(2))

在这些属性约束中, owl:onProperty 用来指出受约束的属性。

4) 本体映射

OWL 提供了一些最简单容易的本体映射功能, 包括声明类和属性的等价、个体的相同或不同等。

a、owl:equivalentClass: 声明类等价;

b、owl:equivalentProperty: 声明属性等价;

c、owl:sameAs: 声明两个个体相同;

d、owl:differentFrom: 声明两个个体不同;

e、owl:AllDifferent: 声明多个两两不同的个体;

f、owl:disjoinWith: 声明两个类不相交。

4.2 模糊本体

在前面的本体章节中, 我们提到本体是“共享概念模型的明确的形式化规范说明”, 其中“明确”是指概念和概念的约束都有明确的和无歧义的定义。那么概念间的关系呢? 难道概念间只有确定的关系(即有这种关系或者没有, 也就是二值逻辑, 非此即彼的关系)? 事实上不是这样的, 好多概念间的关系是模糊的, 即概念间的关系是变化不定的, 需要引入模糊权值 f 来衡量关系的亲密程度。由于本体在表达信息精确上的限制, 我们需要扩展本体的概念, 引入模糊本体。除此之外, 本体中我们定义的概念或者词汇都是相互独立、没有关联的, 而在引入模糊本体的概念之后, 我们定义的概念都是通过别的概念来描述的, 即概念间的关系是密不可分的, 也就是我们所说的信息的互表性。

4.2.1 模糊本体及其形式化

本体层是语义网的核心层, 模糊语义网的核心是模糊本体, 模糊本体是将模糊概念引入到传统的本体模型中, 以解决某一领域的不确定性问题。为了描述概念及概念间关系的不确定性, 本文引入模糊关系, 得到模糊本体模型。本体与模糊本体的差别主要在于模糊权值, 而模糊权值体现在关系程度的描述上。结合本体的定义, 下面给出模糊本体的形式化定义。

定义 6 模糊本体可以定义为一个四元组 $O=(C,R,E,I)$, C 为概念的集合, R 为关系的集合, E 为演化机制(规则), I 为推理机制(应用或解释规则)。 R 中的成员为一个四元组 (r,c_1,c_2,f) , 其中 r 为 c_1 与 c_2 之间的关系名, c_1 、 c_2 为直接相关的两个概念, f 为关系程度, $f \in [0, 1]$, $r, c_1, c_2 \in C$ 。

1) 概念。这里的概念较本体中的概念更为宽泛，可以说是没有概念，因为概念时相互表示的，即一个概念离开和它相关的概念是无法存在的。它可以认为是一个标识，当然也可以是本体中的概念。

2) 关系。关系通常包含定义域和值域两部分，限定了关系所适用的范围。本体中，关系的定义域通常是个概念，值域既可是概念，也可是具体的取值域，当值域为取值域时，关系便退化为属性。若只考虑关系的值域为概念的情况，关系集合 R 中的每个关系 (r, c_1, c_2, f) 便表示概念 c_1 和 c_2 间的二元关系。这个关系只能表明概念 c_1 和 c_2 所对应的实例中可能存在关系 r ，并且关系亲密程度为 f 且 $f \in [0, 1]$ ，当 f 取 0 或者 1 时，此时的模糊本体就退化成了我们前面的本体。层次、实例、公理都可纳入 R 中，是特殊的关系。例如，在概念上的层次有 Kind-of 或 Is-a，表示 Super class-Subclass 关系，我们用 (Kind-of, $c_1, c_2, 0.7$) 表示 c_1 有 70% 的可能是 c_2 的超类。而对于具体的实例我们用 (Is-a, $c_1, c_2, 1$) 表示 c_1 是 c_2 的一个实例。对于公理集合 X 中的每条公理代表领域知识中的永真断言，例如，声明关系 “Teach” 和 “Taught-by” 是互逆的，用关系可以表示为 (Inverse, Teach, Taught-by, 1)。其中关系的程度初始值由专家的经验给出，并不断演化。

3) 演化机制。演化机制贯穿于模糊语义网体系结构中的第三层具体关系描述层 FRDF (S)、第四层抽象关系描述层 Fuzzy Ontology vocabulary、第五层间接关系描述层 Fuzzy Logic。演化机制是事物本身或事物间的关系的变化的规律，我们知道，在大自然界中，事物和事物间的关系都是不断变化的。比如，对于每一个个体人，从出生到死亡，发生了一系列复杂的变化，这只是大自然界中演化（自然演化）中的非常渺小的一个例子。那么在计算机中，我们如何模拟这种变化呢？遵循什么规律呢？这里我们就要提及到演化机制，用来描述计算机中的自然演化规律和机制。

4) 推理机制。推理机制主要应用在模糊语义网体系结构中的语义描述层中具体的见解关系描述层次，给出推理规则，要求按这种规则进行推理。这是自动化智能服务奠定了基础。

4.2.2 模糊本体 Web 语言-FOWL

FOWL 是对 FRDF(S) 的扩展，功能更强的描述语言，可以描述模糊信息。

FRDF(S) 可以表示一些简单的本体，描述某些本体知识，包括子类和子属性关系、属性的定义域和值域约束以及类的实例等。但用作一般的本体表示语言时，FRDF(S) 的表达能力显得太弱，还缺少如下一些特征：

- 1) 对于局部值域的属性定义。
- 2) 类、属性、个体的等价性。
- 3) 不相交类的定义。
- 4) 类的布尔结合定义。
- 5) 基数约束。

6) 关于属性特性的描述。

正是因为 FRDF(S) 具有上述的一些缺陷, 因此需要一种比其描述能力更为丰富的本体语言, 同时要求该语言权衡表达能力和有效的推理支持, 既能支持合理的有效推理, 又能对本体知识充分描述。为此, 我们提出了 FOWL 扩展 FRDF(S), 作为在语义网上表示本体的语言。

1) FOWL 语法概述

FOWL 建立在 FRDF(S) 基础上, 利用了 RDF/XML 语法。FOWL 本体是个 FRDF 文档, 包含在 frdf:RDF 元素中。FOWL 本体以关于本体的声明作为开始。fowl: Ontology 元素用来声明关于当前文档的 FOWL 元数据, 可记录版本信息和导入文档所依赖的定义, 是本体头部的开始。本体头部以 </fowl: Ontology> 结束, 后面跟随的是构成本体的实际定义, 包括类公理、属性公理和个体事实等内容, 最终由 </frdf: RDF> 终止本体定义。

本文中, 我们仅从属性、个体、简单类这三个部分简单介绍 FOWL 语法。

2) 属性、个体和简单类

属性、个体和简单类其实是相互描述相互表示的, 属性是个体的组成部分, 个体是具有相同 domain 的属性的总和, 而类是具有相同或相近属性的个体的集合。

a、属性。属性是个体的组成部分, 是个二元关系。包含定义域和值域两部分, 限定了关系所适用的范围。本体中, 关系的定义域是一个概念, 值域既可是概念, 也可是具体的取值域, 当值域为取值域时, 关系便退化为属性。在 FOWL 中有两种类型的属性: 分别表示类的实例间的关系和类的实例与文字间的关系。

对象属性用 fowl:ObjectProperty 定义, 并可以用 frdfs:domain 和 frdfs:range 指明它的定义域和作用域, frdfs:weight 是属性关系的程度, 还可以将一个属性定义为某个已有属性的子属性。例如:

```
<fowl: ObjectProperty frdfs: ID="createBook">
  <frdfs: subPropertyOf frdfs: resource="create#"/>
    <frdfs: domin frdfs: resource="Person#"/>
    <frdfs: range frdfs: resource="Book#"/>
  <frdfs: weight=0.7/>
</fowl: ObjectProperty>
```

用抽象语法表示如下:

```
ObjectProperty(ex:createBook(ex:weight) super(ex:create)
               domain(ex:Person) range(ex:Book) )
```

其中 super、domain 和 range 都可以出现多次, 指定多个父属性、定义域和值域, 其含义是它们的合取。

属性的使用如下:

```
<fowl: Thing frdfs: ID="DiaryOfTom"/>
```

```
<ex:Man frdfs:ID="Tom"/>
```

```
<ex:createBookfrdfs:resource="#DiaryOfTom" ex:weight(0.7)/>
```

```
</ex:Man>
```

这时可以推断出 DiaryOfTom 一定是本书，即 Book 类的实例，因为 createBook 属性的值域是 Book。用抽象语法中的 value 可以表示为：

```
Individual(ex:Tom type(ex:Man)value(ex:createBook(ex:weight)
ex:DiaryOfTom))
```

其中 value 可以出现多次，说明个体的多个属性。数据类型属性和对象类型的区别在于其值域只能是数据类型。数据类型属性用 fowl:DatatypeProperty 声明，抽象语法中用 DatatypeProperty 定义。

b、个体。个体是具有相同 domain 的属性的总和。事实上，个体除了属性外，我们知道个体还有一些方法或者说是动作或函数，比如说话、做饭、教学、学习等都是一些活动。当然这个我们也可以归结到个体的属性里面，当定义一个具体的个体时，我们不但要给出个体的属性和方法，还需要给出其相应的取值。通俗的讲，我们就是通过给出个体的属性及方法的值才唯一确定下来一个个体的。这里我们对方法不予讨论。下面我们通过一个简单的例子来说明如何用 FOWL 来定义一个个体 Tom:

```
<fowl: Thing frdfs: ID=" Tom"/>
```

```
<frdfs: property rdf: resource="#Age"/>
```

```
<frdfs: weight 1/>
```

```
<frdfs: value 27 />
```

```
<frdfs: property rdf: resource="#Name"/>
```

```
<frdfs: weight 1/>
```

```
<frdfs: value Tom/>
```

```
<frdfs: property rdf: resource="#Gender"/>
```

```
<frdfs: weight 1/>
```

```
<frdfs: value Male/>
```

```
<frdfs: property rdf: resource="#Height"/>
```

```
<frdfs: weight 1/>
```

```
<frdfs: value 178/>
```

```
<frdfs: property rdf: resource="#Weight"/>
```

```
<frdfs: weight 1/>
```

```
<frdfs: value 78/>
```

```
</fowl: Thing>
```

其中 frdfs: property 表示个体的属性，<frdfs: weight 1/>表示个体属性或方法的权值，而<frdfs: value 27 />给出了个体属性的值，

用模糊抽象语法将个体可以表示为:

Individual(ex:Tom property(ex:Age(1)) property (ex: Name(1)))

c、类。具有相同或相近属性的个体的集合。有两个部分组成, 引入或引用一个名称以及一个限制列表(对属性值或权值的限制)。例如, 用 FOWL 来定义一个类青年人 Young:

```
<fowl: Class frdfs:ID="Young">
  <frdfs: property rdf: resource="#Name"/>
  <frdfs: weight 1/>
  <frdfs: value String/>
  <frdfs: property rdf: resource="#Gender"/>
  <frdfs: weight 1/>
  <frdfs: value (Male,Female)>
  <frdfs: property rdf: resource="#Age"/>
  <frdfs: weight 1/>
  <frdfs: value (18,45)/>
  <frdfs: property rdf: resource="#Count"/>
  <frdfs: weight 1/>
  <frdfs: value 0/>
</fowl: Class>
```

其中, <frdfs: property rdf: resource="#Name"/>代表类的属性姓名;

<frdfs: property rdf: resource="#Gender"/>代表类的属性性别; <frdfs: property rdf: resource="#Age"/>代表类的属性年龄; <frdfs: property rdf: resource="#Count"/>代表类的属性数量, 用于统计属于该类的成员个数。

对于一个确定的类, 其不但要求有属性, 而且还必须给出这些属性的取值范围, 只有这样才能对其进行限定。对于缺省权值和限制的属性, 表示对该属性不加任何限制。

假设该本体的名称空间为 ex, 对应的抽象语法可以表示为:

```
Class (ex: Young property (ex: Age (18, 45)) property (ex: Name) property (ex: Gender)
property (ex: Count))
```

有了以上的模糊本体 Web 语言-FOWL 的定义, 我们就可以用 FOWL 表示模糊语义网了。

4.2.3 用 FOWL 表示模糊本体

在模糊资源描述框架思想的基础上, 提出模糊语义网的设想: 模糊语义网是对当前 Web 的改进, 它将 Web 上的资源赋予模糊性, 描述了资源间的模糊关系, 而且包含语义信息, 以便于机器的自动处理。

将图 3.5 用 RDF/XML 语法表示, 如下:

```

<? xml version="1.0"?>
<rdf:RDF xmlns:rdf=http://www.w3.org/1999/02/22-rdf-syntax-ns#
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:ex="http://www.example.org/">
  <rdf:Description rdf:about="http://www.example.org/index.html">
    <dc:creator ex:weight=0.8 rdf:resource="http://example.org/people1023"
      ex:name="Tom" ex:weight =0.9
ex:email="Tom@email.org" ex:weight =0.7/>
  </rdf:Description>
  <rdf:Description rdf:about="http://www.example.org/main.html">
    <dc:creator ex:weight =0.5 rdf:resource="http://example.org/people1023"
      ex:name="Tom" ex:weight =0.9
ex:email="Tom@email.org" ex:weight =0.7/>
  </rdf:Description>
</rdf:RDF>

```

上述 RDF 代码采用 ex:weight 来描述主体与客体之间具有某种关系的可信度,即表示权值的大小。

用 FOWL 语法描述如下:

```

<? xml version="1.0"?>
<rdf:RDF xmlns:rdf=http://www.w3.org/1999/02/22-rdf-syntax-ns#
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:ex="http://www.example.org/">
  <fowl: Thing rdf:about="http://www.example.org/index.html">
    <frdfs: property rdf: resource="#creator"/>
    <frdfs: value rdf: resource=" http://example.org/people1023"/>
    <frdfs: weight 0.8/>
    <frdfs: property rdf: resource="#name"/>
    <frdfs: value rdf: resource="Tom"/>
    <frdfs: weight 0.9/>
    <frdfs: property rdf: resource="#email"/>
    <frdfs: value rdf: resource=" Tom@email.org"/>
    <frdfs: weight 0.7/>
  </fowl:Thing>
  <fowl: Thing rdf:about="http://www.example.org/main.html">
    <frdfs: property rdf: resource="#creator"/>

```

```
<frdfs: value rdf: resource=" http://example.org/people1023"/>
<frdfs: weight 0.5/>
<frdfs: property rdf: resource="#name"/>
<frdfs: value rdf: resource="Tom"/>
<frdfs: weight 0.9/>
<frdfs: property rdf: resource="#email"/>
<frdfs: value rdf: resource=" Tom@email.org"/>
<frdfs: weight 0.7/>
</fowl:Thing>
</rdf: RDF>
```

5 模糊语义网的演化研究

自然界中的演化从来就没停止过，具有普遍性。比如一个个体人，从出生到死亡，所经历的一切如年龄的增长、头发的变白、思维的变化、知识的增长、身高体重的增长都可看做自然演化。还比如一个国家的发展、人民的生活水平、国际关系等都是时刻发生着变化、这些变化或微妙或巨大，然而这些只是大自然演化中的冰山一角。模糊语义网不但要描述这些信息，还要描述这些信息的变化，从而更客观地描述现实世界。

模糊语义网体系结构中的模糊本体层用来描述资源以及定义不同概念间的关系，以支持词汇的演化。因此，通常对模糊语义网的演化研究实质上就是针对语义描述层的演化研究。而语义描述层的演化研究又包括具体关系的演化、抽象关系的演化以及间接关系的演化。这里我们只讨论具体关系的演化和抽象关系的演化。

5.1 具体关系的演化研究

5.1.1 反馈神经网络

反馈神经网络是拓扑结构中有环路的神经网络。最著名的是 Hopfield 神经网络，有两种形式，（连续型和离散型）。离散 Hopfield 网络结构示意图如图 5.1 所示。

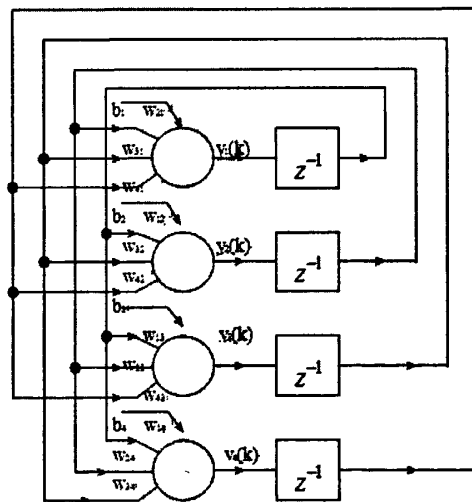


图 5.1 离散 Hopfield 网络模型

Hopfield 网络为一层结构的反馈网络，能处理双极型离散数据（即输入-1，+1），及二进制数据（0，1）。当网络经过适当训练后，可以认为网络处于等待的工作状态。而对网络给定初始输入时，网络就处于特定的初始状态。由此初始状态开始运行，可得到网络输出。然后这个输出状态通过反馈连接会送到网络的输入端，作为网络下一阶段运行的输入信号，该输入信号可能与输入信号不同。由于这个新的输入可得到下一步的输出，该输

出也可能与上一步的输出不同。如此下去,网络的整个运行过程就是上述反馈过程的重复。如果网络是稳定的,那么随着多次反馈运行,网络状态的变化逐渐减少,最后不再变化,达到稳定。

离散和连续 Hopfield 网络都有从初始状态动态收敛到稳定状态的能力,充分利用这个能力可以使 Hopfield 网络实现优化计算和联想记忆等功能。连续型 Hopfield 网络主要用于优化计算,离散型 Hopfield 网络则主要用于联想记忆。

5.1.2 具体关系的演化算法

本体演化是指一个本体面对外部环境变化要能够具备随时间变化的适应能力^[38]。随着时间推移,由于知识具有内在的变化性质,具有高度的动态性质;领域的变化,对不同任务的适应性,以及本体在概念化和关系上的变化均要求给出的知识表示方法要具备良好的动态适应性。引起本体演化的三个主要因素如下:

1) 领域的变化。领域的变化非常普遍,有些变化是隐式的,只有在对用户与系统进行交互分析后才能发现。

2) 共享概念模型的变化。概念语义的改变,即上下文语义的变化,是随着领域视图或使用角度的改变引起的,

3) 表示的变化。表示的变化是指种转换,本体由一种知识表示语言用另一种语言表示时,会产生显式改变。

本文在学习本体的知识之后,研究模糊语义网的演化,本体的概念在前面章节已经介绍过了。本体是“共享概念模型的明确的形式化规范说明”。但是,在日常生活中常常会遇到对某些概念的理解不是很确定,存在模糊性。而且这种模糊程度还会随着时间及空间等因素的影响而发生变化。这就是模糊语义网的演化。在创建本体的初期,领域专家根据以前的事实经验及对当前事实的考察定义了领域范围内的本体。但是,随着社会的进步和科技的发展,每天都有新的事物诞生,这就使得本体也发生了改变。可能是添加了某些新的概念,也可能是旧的概念被新的概念所取代,还有可能原来关系很微小的两个概念之间的关系明朗化。这就需要领域专家能够及时地修改本体内容,还要能够根据历史的走向预知未来的发展趋势。

根据反馈神经网络的思想,将本体中的概念看成是网络上的节点,演化的过程可看成是根据节点间连接的变化情况确定节点的语义。任意两个节点间都有连接权值,表示了节点之间的连接强度。如图 5.2 所示,为基于反馈神经网络的演化示意图。

图中的 P 是反馈神经网络的输入,代表用户当前关注的概念; Σ 是神经网络的神经元,作用是把用户层的概念转成本体层的概念;F 也是神经网络的神经元,它代表的是与输入概念相关联的概念;Y 是神经网络的输出,也是用户概念语义的输出; ω_i 是输入 P 与中间结点 F 的连接权值,代表了用户当前关注的概念和与其关联的概念间的关联程度; λ 是连接系数。

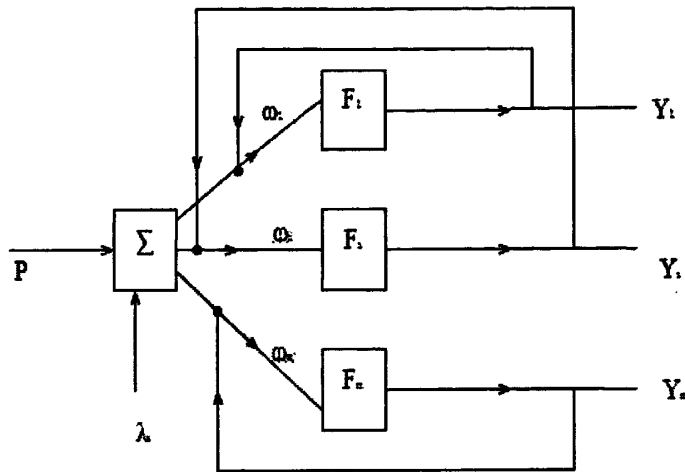


图 5.2 演化示意图

输入节点与每个与其相关的节点之间都有连接权值，相应的权值代表了关注节点和与其相关节点的关联程度，权值的变化表示了节点本身意义的变化。神经网络的输出是对输入节点的一种解释，每次输出对下次权值变化提供了历史依据，那么权值的改变是历史积累的，历史积累就是演变。

根据对反馈型神经网络的研究，提出本体的演化算法：

$$\omega_t(i) = (1 + \lambda_{t-1}(i)) \omega_{t-1}(i) \quad (5.1)$$

其中， ω_t 为第 t 次迭代时的权值； $\lambda(i)$ 为主体 A 与第 i 个客体 B_i 间的连接常系数。 λ 的初值为相同的较小正数，其大小由网络的规模来定。网络规模大， λ 的取值就小；网络规模小，则 λ 取值就大些。并且在演化过程中， λ 的取值不变。

演化过程中，根据用户的偏好选择，选中输出与用户输入节点间的连接权值按公式 5.1 演变，其他与用户输入节点相关的连接权值不发生变化。一次演化过程结束后，权值要归一化，归一化公式如下：

$$\omega_t(i) = \omega_t(i) / \sum_{j=0}^n \omega_t(j) \quad (5.2)$$

归一化的目的是使每次演化后相关权值的总和为 1，权值的大小表示关联强度的大小。这样，虽然演化算法每次只改变用户选择的偏好，但是，经过归一化后，与用户当前输入有关的连接权值都发生了演变。

举个例子来说明图 5.2 的理论，用户当前关注的“苹果”一词作为反馈神经网络的输入，输入到神经元 Σ ，神经元 Σ 根据历史依据进行分析推理，确定用户输入的“苹果”一词的语义是“水果”、“数码”还是“服装”，假设用户的语义是“数码”，那么利用

公式(5.1)来改变“苹果”与“数码”之间的连接权值,而“苹果”与“水果”及“服装”之间的连接权值不变;然后根据归一化公式(5.2)进行归一处理,并将修改后的权值反馈给神经元;最后输出当前用户输入的最佳语义。

5.2 抽象关系的演化研究

具体关系的变化导致了抽象关系的改变^[39],比如在我们的职称晋升本体实验中,由于个体人的工作量或科研量等属性发生了变化,最终导致了个体属于其它的类。那么研究完了具体关系的演化规律,我们很有必要研究一下抽象关系的演化到底是如何的?

5.2.1 抽象关系的演化算法

所谓演化就是指概念本身或其属性受到外界或自身内在的变化而使得概念或概念间的关系发生了变化。在职称晋升本体演化实验中,我们模拟随机改变个体的4个属性(岗位年限 workTime、教学量 workload、科研量 research、其他 other)值之一。通过下面的方法实现:

定义一个1-4的随机数 style:

```
int style = (int)(Math.random()*5);
```

如果 style=4, 改变岗位年限;

```
if(style==4)
```

```
{updateXMLworkTime(workTime_nl.item(number).getFirstChild());}
```

如果 style=3, 改变教学量;

```
if(style==3)
```

```
{updateXMLworkload(workload_nl.item(number).getFirstChild());}
```

如果 style=2, 改变科研量;

```
if(style==2)
```

```
{updateXMLresearch(research_nl.item(number).getFirstChild());}
```

如果 style=1, 改变其他;

```
if(style==1)
```

```
{updateXMLother(other_nl.item(number).getFirstChild());}
```

具体每一项的值按照以下规则进行变化:

修改岗位年限 workTime,如果原来是 0-3 年 (5 分), 经过一次演化就变为 3-5 年 (10 分), 如果是 3-5 年那就就变为 5 年以上 (15 分)。

```
public static boolean updateXMLworkTime(Node node)
```

```
{
```

```
    boolean re = false;
```

```
    if(node.getNodeValue().equals("5"))
```

```

        node.setNodeValue("10");
    else
        if(node.getNodeValue().equals("10"))
            node.setNodeValue("15");
        re = true;
        return re;
    }

```

修改教学量 workload, 如果任务量小于 30, 每次演化自增 2, 如果增 2 后大于 30, 那
改为 30。

```

public static boolean updateXMLworkload(Node node)
{
    boolean re = false;
    long temp = Long.parseLong(sss.getNodeValue());
    if(temp < 30)
    {
        temp = temp + 2;
        if(temp > 30)
            temp = 30;
    }
    node.setNodeValue("" + temp);
    re = true;
    return re;
}

```

修改科研量 research, 如果科研量小于 40, 每次演化自增 2, 如果增 2 后大于 40, 那
改为 40。

```

public static boolean updateXMLresearch(Node node)
{
    boolean re = false;
    long temp = Long.parseLong(node.getNodeValue());
    if(temp < 40)
    {
        temp = temp + 2;
        if(temp > 40)
            temp = 40;
    }
}

```

```

node.setNodeValue(""+temp);
re = true;
return re;
}

```

修改其他项 other, 如果其他小于 15, 每次演化自增 1。

```

public static boolean updateXMLother(Node node)
{
    boolean re = false;
    long temp = Long.parseLong(node.getNodeValue());
    if(temp<10)
    {
        temp = temp+1;
    }
    node.setNodeValue(""+temp);
    re = true;
    return re;
}

```

以上就是个体属性演化的算法说明, 下面我们考虑如何计算个体对于类的隶属度, 也就是抽象关系的演化算法。

5.2.2 个体隶属度的计算方法

当我们定义好了一些个体及类后, 如何判断某个个体对于某个类的隶属度? 直白的说, 就是要判断一个个体是否属于某个类, 如果不完全属于这个类, 那么属于这个类的程度是多少(我们用一个百分比表示, 比如 78%, 我们认为这个个体隶属于或依赖于某个类的程度是 78%)? 我们把这个程度叫做隶属度。下面我们给出隶属度的算法:

设我们有一个个体 E, 该个体有 m 个属性, 每个属性有不同的权值, 我们用 $E(p_1, p_2, p_3, \dots, p_m)$ 表示, 其中 p_i 为 (pw_i, pv_i) (其中 p:property 属性; pw: property weight 属性权值; pv: property value 属性值)。

而对于每个类 Class, 它都有固定的属性, 因为类是有差别的, 在类的定义中, 类的差别主要体现在类的属性个数和对属性的约束上。我们用 $Class(cp_1, cp_2, cp_3, \dots, cp_n)$ 表示, cp_i 为 (cw_i, cv_i) (其中 cp:class property 类属性; cw: class weight 类属性权值; cv: class value 类属性限制的值)。有了以上的定义, 我们现在给出隶属度算法(前提是属性个数相同, 即 $m \geq n$):

$$p = \frac{1}{n} \sum_{i=1}^n f(pw_i, cw_i) * g(pv_i, cv_i) \quad (5.4)$$

其中, n 代表类属性的个数(记住是类不是个体), pw_i 表示个体属性的权值, cw_i 表示类属性的权值, pv_i 表示各个属性的值, cv_i 表示属性的类属性的值域取值范围。我们之所以给出这样的算法, 是因为我们通过研究发现, 隶属度和 pw 、 cw 、 pv 、 cv 这四个属性有关, 但具体是什么关系要根据具体的情况给出判断。下面我们需要分情况予以讨论:

1) 描述两个概念之间的关系

举例如下:

我们给出一些概念, 分别是:

水果类: 包括名字、重量、颜色、产地等属性。

```
<fowl: Class frdfs: ID="Apple"/>
```

```
<frdfs: property rdf: resource="#Name"/>
```

```
<frdfs: weight cw1/>
```

```
<frdfs: value 水果/>
```

```
<frdfs: property rdf: resource="#Weight"/>
```

```
<frdfs: weight cw2/>
```

```
<frdfs: value 0.5 />
```

```
<frdfs: property rdf: resource="#Color"/>
```

```
<frdfs: weight cw3/>
```

```
<frdfs: value 红色 />
```

```
<frdfs: property rdf: resource="#Place"/>
```

```
<frdfs: weight cw4/>
```

```
<frdfs: value 陕西富平 />
```

```
</fowl: Class >
```

苹果品牌服装类: 包括名字、重量、颜色、产地等属性。

```
<fowl: Class frdfs: ID=" Apple"/>
```

```
<frdfs: property rdf: resource="#Name"/>
```

```
<frdfs: weight cw2/>
```

```
<frdfs: value 服装品牌 />
```

```
<frdfs: property rdf: resource="#Weight"/>
```

```
<frdfs: weight cw2/>
```

```
<frdfs: value 0.5 />
```

```
<frdfs: property rdf: resource="#Color"/>
```

```
<frdfs: weight cw3/>
```

```
<frdfs: value 粉红色 />
```

```
<frdfs: property rdf: resource="#Place"/>
```

```
<frdfs: weight cw4/>
```

```

<frdfs: value 浙江温州 />
</fowl: Class >
苹果电器类: 包括名字、重量、颜色、产地等属性。
<fowl: Class frdfs: ID=" Apple"/>
<frdfs: property rdf: resource="#Name"/>
<frdfs: weight cw3/>
<frdfs: value 苹果电器 />
<frdfs: property rdf: resource="#Weight"/>
<frdfs: weight cw2/>
<frdfs: value 2 />
<frdfs: property rdf: resource="#Color"/>
<frdfs: weight cw3/>
<frdfs: value 天蓝色 />
<frdfs: property rdf: resource="#Place"/>
<frdfs: weight cw4/>
<frdfs: value 美国硅谷 />
</fowl: Class >

```

这里的苹果我们涉及到三个概念, 分别指水果、服装和电器, 这里的类权值 *cwi* 表示该类是否有名字, 所以我们应当取值为 1。那么当我们谈及苹果这个抽象概念时, 我们一般会想到什么? 或者说苹果这个抽象概念和我们这三个具体的概念之间的权值是怎样的?

苹果:

```

<fowl: Class frdfs: ID="Apple">
<frdfs: property rdf: resource="#Name"/>
<frdfs: weight cw1/>
<frdfs: value Apple />
<frdfs: property rdf: resource="#Weight"/>
<frdfs: weight cw2/>
<frdfs: value 0.2 />
<frdfs: property rdf: resource="#Color"/>
<frdfs: weight cw3/>
<frdfs: value 颜色 />
<frdfs: property rdf: resource="#Place"/>
<frdfs: weight cw4/>

```

<frdfs: value 产地 />

</fowl: Class>

下面我们给出分析:

这种情况下, 每个具体概念的属性权值均为 1, 表示每个具体概念都有名字、重量、颜色、产地等这些属性, 而抽象概念中的权值 cw_1 则表示了具体概念(水果、苹果品牌服装、苹果电器)的名字属性在抽象概念(苹果)名字中的权重。就如上面的例子, 当考虑到抽象概念苹果和三个具体概念(水果、苹果品牌服装、苹果电器)的关系时, 那么这三个概念和抽象概念的权值总和应为 1。也就是说, 我们提到的苹果这个概念, 必属于以上三个具体概念之一。只不过在不同领域, 面对不同的主体, 他们的权值有所差别。例如, 在电脑市场, 有可能是这样的结果:

苹果(电器)——苹果 $cw_1=0.7$

苹果(服装)——苹果 $cw_2=0.2$

苹果(水果)——苹果 $cw_3=0.1$

而在服装批发市场, 可能会有这样的结果:

苹果(电器)——苹果 $cw_1=0.1$

苹果(服装)——苹果 $cw_2=0.8$

苹果(水果)——苹果 $cw_3=0.1$

那么, 如果在水果批发市场, 以下的权重会更合理些

苹果(电器)——苹果 $cw_1=0.1$

苹果(服装)——苹果 $cw_2=0.1$

苹果(水果)——苹果 $cw_3=0.8$

那么对于大学生这个领域而言, 服装、电器、水果的比重可能相差不大, 比如

苹果(电器)——苹果 $cw_1=0.4$

苹果(服装)——苹果 $cw_2=0.3$

苹果(水果)——苹果 $cw_3=0.3$

事实上, 这些权值会随着使用者的偏好有所变化, 这是由于每个终端作用的结果, 也就是说, 权值会随着大众的选择而变化。

这里的抽象概念类属性 cw_1 权值是不给出的, 它表示的是某个具体概念的该属性在抽象概念中的属性的权值。

下面我们给出这种情况下的隶属度:

$$p = \frac{1}{n} \sum_{i=1}^n f(pw_i, cw_i) * g(pv_i, cv_i) = \frac{1}{1} f(1, cw) * 1 = \frac{cw}{1} = cw \quad (5.5)$$

其中, n 代表类属性的个数(记住是类不是个体), pw_i 表示个体属性的权值, cw_i 表示类属性的权值, pv_i 表示各个属性的值, cv_i 表示属性的类属性的值域取值范围。

也就是说, 这种情况下, 某个具体概念对于抽象概念的隶属度其实就是抽象概念的属

性权值，当然这个权值的值是多少，和具体领域有关，最终由领域专家根据经验给出。

2) 当我们需要将人类进行婴儿、少儿、少年、青年、中年、老年这样的划分时，此时的隶属度算法有待研究

我们对各个年龄阶段的划分见表 5.1:

表 5.1 年龄划分表

名称	婴儿	少儿	少年	青年	中年	老年
年龄段	1-3 岁	4-10 岁	10-18 岁	18-45 岁	46-60 岁	>60 岁

我们知道这样的划分是通过年龄划分，那么我们就主要拿年龄这个属性进行讨论。首先给出个体 Tom 和 Jerry 以及青年类 Young。

Tom:

```
<fowl: Thing frdfs: ID="Tom"/>
<frdfs: type rdf: resource="#Person"/>
<frdfs: property rdf: resource="#Age"/>
<frdfs: weight 1/>
<frdfs: value 32 />
<frdfs: property rdf: resource="#Name"/>
<frdfs: weight 1/>
<frdfs: value Tom />
<frdfs: property rdf: resource="#Gender"/>
<frdfs: weight 1/>
<frdfs: value Male />
<frdfs: property rdf: resource="#Height"/>
<frdfs: weight 1/>
<frdfs: value 167 />
</fowl: Thing>
```

Jerry:

```
<fowl: Thing frdfs: ID="Jerry"/>
<frdfs: type rdf: resource="#Person"/>
<frdfs: property rdf: resource="#Age"/>
<frdfs: weight 1/>
<frdfs: value 19 />
<frdfs: property rdf: resource="#Name"/>
```

```

<frdfs: weight 1/>
<frdfs: value Jerry />
<frdfs: property rdf: resource="#Gender"/>
<frdfs: weight 1/>
<frdfs: value Female />
<frdfs: property rdf: resource="#Height"/>
<frdfs: weight 1/>
<frdfs: value 158 />
</fowl: Thing>
青年类 Young:
<fowl: Class frdfs: ID="Young">
<frdfs: property rdf: resource="#Age"/>
<frdfs: weight cw/>
<frdfs: value (18, 45) />
<frdfs: property rdf: resource="#Name"/>
<frdfs: weight 1/>
<frdfs: value 青年 />
<frdfs: property rdf: resource="#Gender"/>
<frdfs: weight 1/>
<frdfs: value all />
<frdfs: property rdf: resource="#Height"/>
<frdfs: weight 1/>
<frdfs: value (0-260) />
</fowl: Class>

```

我们看到, Tom 和 Jerry 的年龄属性权值均为 1, 表示他们有该属性, 我们可以看出, 这两个个体都属于青年类, 那么我们如何计算出他们的青年度呢? (注意不是年轻度), 这里的青年类中的 cw 就表示青年度。在 (18,45) 这个年龄区间, 越靠近中间的青年度是越大的, 而不是年龄越小的青年度越大, 初步分析认为, 这是一个符合正态分布的走势。

标准正态分布函数为:

$$p(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad (-\infty < x < +\infty) \quad (5.6)$$

我们这里基本可以认为符合 $\mu=(18+45)/2=31.5$ $\sigma=1$ 的正态分布;

我们定义, 当 $\text{age} \leq (1/2) * (18+45)$ 时, 隶属度

$$p(age) = 2 \frac{\int_{18}^{age} p(x) dx}{\int_{18}^{45} p(x) dx}; \quad (5.7)$$

当 $age \geq (1/2) * (18+45)$ 时,

$$p(age) = 2 \left(\frac{\int_{18}^{45} p(x) dx}{\int_{18}^{age} p(x) dx} \right). \quad (5.8)$$

我们可以计算出 Tom 的青年度是

$$p(32) = 2 \frac{\int_{18}^{32} p(x) dx}{\int_{18}^{45} p(x) dx} = 2 \frac{\Phi(32) - \Phi(18)}{\Phi(45) - \Phi(18)}; \quad (5.9)$$

Tom 的青年度是

$$p(19) = 2 \frac{\int_{18}^{19} p(x) dx}{\int_{18}^{45} p(x) dx} = 2 \frac{\Phi(19) - \Phi(18)}{\Phi(45) - \Phi(18)}. \quad (5.10)$$

其中 $\Phi(x)$ 是 $p(x)$ 的原函数。

3) 评职称 (助教 讲师 副教授 教授) 或学位 (初中 高中 学士 硕士 博士) 类似这样的问题

我们都知道量变引起质变, 也就是说变化都有一个过渡的过程, 比如从副教授到教授, 可能随着论文的发表不断积分, 达到一定程度可以评为教授, 而不是我们看到的离散的, 跳跃的过程, 我们现在就要描述出这个渐变的过程。

下面我们给出一个个体 Tom, 以及三个学位类 (学士 硕士 博士);

Tom:

```
<fowl: Thing frdfs: ID="Tom"/>
<frdfs: type rdf: resource="#Person"/>
<frdfs: property rdf: resource="#Name"/>
<frdfs: weight 1/>
<frdfs: value Tom />
<frdfs: property rdf: resource="#Gender"/>
<frdfs: weight 1/>
<frdfs: value Male />
```

```

<frdfs: property rdf: resource="#Age"/>
<frdfs: weight 1/>
<frdfs: value 27 />
<frdfs: property rdf: resource="#Degree1"/>
<frdfs: weight 1/>
<frdfs: value Bachelor />
<frdfs: property rdf: resource="#Degree2"/>
<frdfs: weight 0.8/>
<frdfs: value Master/>
<frdfs: property rdf: resource="#Degree3"/>
<frdfs: weight 0.1/>
<frdfs: value Dr./>
</fowl: Thing>
学士类:
<fowl: Class frdfs: ID="Bachelor">
<frdfs: property rdf: resource="#Degree"/>
<frdfs: weight 1/>
<frdfs: value Bachelor Degree />
<frdfs: property rdf: resource="#Name"/>
<frdfs: weight 1/>
<frdfs: value 学士 />
<frdfs: property rdf: resource="#Deadline"/>
<frdfs: weight 1/>
<frdfs: value 4 />
<frdfs: property rdf: resource="#School"/>
<frdfs: weight 1/>
<frdfs: value 学校 />
<frdfs: property rdf: resource="#DegreeType"/>
<frdfs: weight 1/>
<frdfs: value (工学, 理学) />
</fowl: Class>
硕士类:
<fowl: Class frdfs: ID="Master">
<frdfs: property rdf: resource="#Degree"/>
<frdfs: weight 1/>

```

```

<frdfs: value Master Degree />
<frdfs: property rdf: resource="#Name"/>
<frdfs: weight 1/>
<frdfs: value 硕士 />
<frdfs: property rdf: resource="#Deadline"/>
<frdfs: weight 1/>
<frdfs: value 3 />
<frdfs: property rdf: resource="#School"/>
<frdfs: weight 1/>
<frdfs: value 学校 />
<frdfs: property rdf: resource="#DegreeType"/>
<frdfs: weight 1/>
<frdfs: value (工学, 理学) />
</fowl: Class>
博士类:
<fowl: Class frdfs: ID=" Dr.">
<frdfs: property rdf: resource="#Degree"/>
<frdfs: weight 1/>
<frdfs: value Dr.Degree />
<frdfs: property rdf: resource="#Name"/>
<frdfs: weight 1/>
<frdfs: value 博士 />
<frdfs: property rdf: resource="#Deadline"/>
<frdfs: weight 1/>
<frdfs: value (3, 8) />
<frdfs: property rdf: resource="#School"/>
<frdfs: weight 1/>
<frdfs: value 学校 />
<frdfs: property rdf: resource="#DegreeType"/>
<frdfs: weight 1/>
<frdfs: value (工学, 理学) />
</fowl: Class>

```

通过以上的描述, 我们可以看到, 个体 Tom 除了姓名、性别、年龄属性外, 还有 3 个学位属性, 分别是学士、硕士、博士。其权值分别是 $pw1=1, pw2=0.8, pw3=0.1$, 表示了

获得学位的程度。 $pw_1=1$ 表示已获得学士学位，而 $pw_2=0.8$ 表明硕士在读，或者是研二，经过答辩或者随着论文的发表就可以毕业拿到硕士学位，总之是比较接近硕士学位，那么 $pw_3=0.1$ 则表示了有考博的想法等。当然这些初始值不是由我们自己给出，要经过权威专家根据一些指标来给出，当然这些权值会随着论文的发表、考试、或者答辩之类的活动后有所变化，也就是我们所说的演化。而学士、硕士、博士类中的学位这一权值 cw 均为 1，表明必须取得该学位后才能称为学士、硕士或博士，也就是我们通常所关注的质变。

那么，在这种情况下，

Tom 对于学士的隶属度是：

$$p_1 = \frac{1}{n} \sum_{i=1}^n f(pw_i, cw_i) * g(pv_i, cv_i) = \frac{1}{1} * f(pw_1, cw) * 1 = \frac{pw_1}{cw} = \frac{pw_1}{1} = pw_1 = 1; \quad (5.11)$$

Tom 对于硕士的隶属度是：

$$p_2 = \frac{1}{n} \sum_{i=1}^n f(pw_i, cw_i) * g(pv_i, cv_i) = \frac{1}{1} * f(pw_2, cw) * 1 = \frac{pw_2}{cw} = \frac{pw_2}{1} = pw_2 = 0.8; \quad (5.12)$$

Tom 对于博士的隶属度是：

$$p_3 = \frac{1}{n} \sum_{i=1}^n f(pw_i, cw_i) * g(pv_i, cv_i) = \frac{1}{1} * f(pw_3, cw) * 1 = \frac{pw_3}{cw} = \frac{pw_3}{1} = pw_3 = 0.1; \quad (5.13)$$

6 职称晋升本体及其演化实验

6.1 实验系统的设计及实现

6.1.1 系统功能分析

本实验系统由 2 个模块组成，即个体添加、个体演化。两个模块的具体表现为：

1) 个体添加。对于新的个体，用户可以通过个体添加组件将其添加的个体库中。

2) 演化。用户选择了演化的个体后，系统会根据用户的选择，采用模拟随机演化算法，随机的改变改个体的某个属性值，并保存在 XML 数据库中。每次演化后个体的分值都不一样，因此对于不同类的隶属度也有所变化。

6.1.2 系统结构的设计

本实验系统的结构采用 MVC 设计模式，即 Model-View-Controller 模式，核心思想是将程序代码分成相对独立又协同工作的三部分。具体功能如下所示^[34]：

模型（Model）：业务逻辑层。实现具体状态管理、业务逻辑的功能。

视图（View）：表示层。与用户实现交互，通常实现数据输入、输出功能。

控制器（Controller）：控制层。控制整个业务流程，实现 View 和 Model 协同工作。

MVC 设计模式的结构及各组成部分间的通信方式如图 6.1 所示：

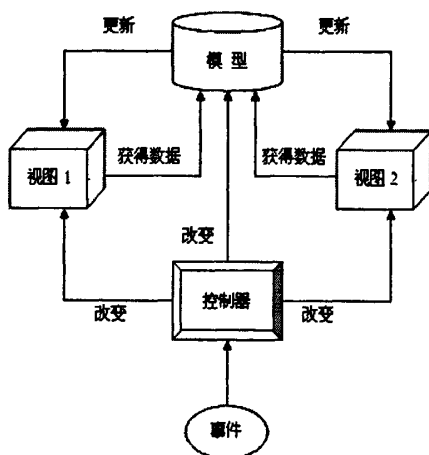


图 6.1 MVC 设计模式结构

本实验采用 Servlet 作为控制器，JavaBean 作为模型，JSP 作为视图。Servlet 负责接收客户端 Web 浏览器发送来的请求，依据不同的请求调用不同的 JavaBean 处理具体的业务逻辑，然后 Servlet 再依据处理的不同结果，转发到对应的 JSP 页面，实现在浏览器客

户端的显示。具体实现方式如图 6.2 所示：

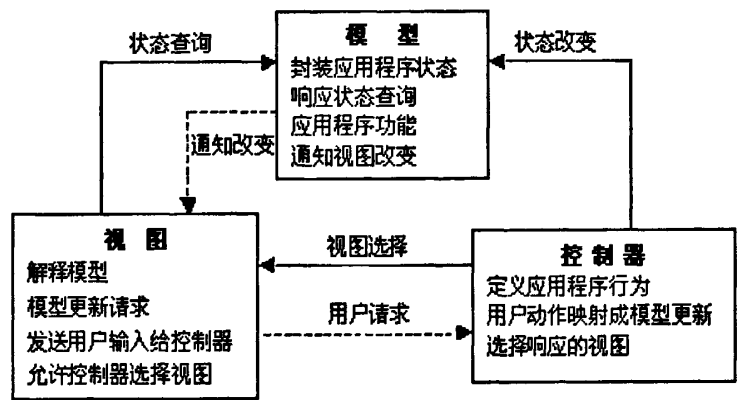


图 6.2 实验设计模式

6.1.3 系统流程

演化是实验系统的主体部分，演化流程图如图 6.3 所示。

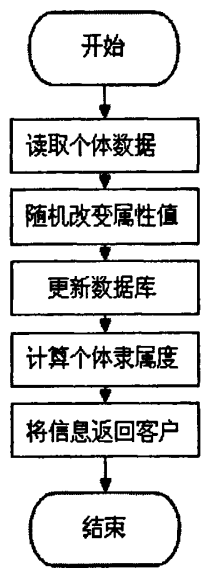


图 6.3 演化流程

系统首先获取个体的信息，展现到客户端。其次，系统获取用户的选择信息，根据用户的选择个体，随机改变该个体的四个属性之一的值，更新个体库中个体的属性值。并将新产生的值存入 XML 数据库中，以备下次演化使用。至此，完成了一次演化。在多次演

化过程中,个体与类的隶属度发生着改变,隶属度的这种改变就是演化的体现,它使得信息平滑的过渡、描述变得更加准确。

6.1.4 系统实现步骤

1) 创建类

本实验系统涉及到4个描述岗位类,分别是助教、讲师、副教授、教授。本实验的前提是所有的个体均为教师,我们在描述这四个类的时候只关注了该岗位所需要的达到的分数这一属性限制,其他的属性限制省略了。

下面我们用 FOWL 语法给出类的描述:

助教类:

```
<? xml version="1.0" encoding="UTF-8"?>
<temp>
<fowl: Class>
<frdfs: name>AssistantTeachers</frdfs: name>
<frdfs: workTime>2</frdfs: workTime>
<frdfs: workLoad>10</frdfs: workLoad>
<frdfs: research>20</frdfs: research>
<frdfs: other>5</frdfs: other >
<frdfs: mark>60</frdfs: mark >
</fowl: Class>
</temp>
```

讲师类:

```
<? xml version="1.0" encoding="UTF-8"?>
<temp>
<fowl: Class>
<frdfs: name>Lecturer</frdfs: name>
<frdfs: workTime>3</frdfs: workTime>
<frdfs: workLoad>15</frdfs: workLoad>
<frdfs: research>25</frdfs: research>
<frdfs: other>10</frdfs: other >
<frdfs: mark>70</frdfs: mark >
</fowl: Class>
</temp>
```

副教授类:

```
<? xml version="1.0" encoding="UTF-8"?>
```

```

<temp>
<fowl: Class>
<frdfs: name>ViceProfessor</frdfs: name>
<frdfs: workTime>4</frdfs: workTime>
<frdfs: workLoad>20</frdfs: workLoad>
<frdfs: research>30</frdfs: research>
<frdfs: other>12</frdfs: other >
<frdfs: mark>80</frdfs: mark >
</fowl: Class>

```

```

</temp>

```

教授类:

```

<? xml version="1.0" encoding="UTF-8"?>

```

```

<temp>
<fowl: Class>
<frdfs: name>Professor</frdfs: name>
<frdfs: workTime>5</frdfs: workTime>
<frdfs: workLoad>25</frdfs: workLoad>
<frdfs: research>35</frdfs: research>
<frdfs: other>15</frdfs: other >
<frdfs: mark>90</frdfs: mark >
</fowl: Class>

```

```

</temp>

```

将这四个类存储在一个 Class.xml 文件中, 如下:

```

<? xml version="1.0" encoding="UTF-8"?>
<temp>
<fowl: Class>
<frdfs: name>AssistantTeachers</frdfs: name>
<frdfs: workTime>2</frdfs: workTime>
<frdfs: workLoad>10</frdfs: workLoad>
<frdfs: research>20</frdfs: research>
<frdfs: other>5</frdfs: other >
<frdfs: mark>40</frdfs: mark >
</fowl: Class>
<fowl: Class>
<frdfs: name>Lecturer</frdfs: name>

```

```

<frdfs: workTime>3</frdfs: workTime>
<frdfs: workLoad>15</frdfs: workLoad>
<frdfs: research>25</frdfs: research>
<frdfs: other>10</frdfs: other >
<frdfs: mark>50</frdfs: mark >
</fowl: Class>
<fowl: Class>
<frdfs: name>ViceProfessor</frdfs: name>
<frdfs: workTime>4</frdfs: workTime>
<frdfs: workLoad>20</frdfs: workLoad>
<frdfs: research>30</frdfs: research>
<frdfs: other>12</frdfs: other >
<frdfs: mark>60</frdfs: mark >
</fowl: Class>
<fowl: Class>
<frdfs: name>Professor</frdfs: name>
<frdfs: workTime>5</frdfs: workTime>
<frdfs: workLoad>25</frdfs: workLoad>
<frdfs: research>35</frdfs: research>
<frdfs: other>15</frdfs: other >
<frdfs: mark>70</frdfs: mark >
</fowl: Class>
</temp>

```

2) 创建个体

本试验程序的个体可以是动态添加的, 个体的前提是教师, 每个个体我们挑出了最关注的 5 个属性。分别岗位 education、岗位年限 worktime、教学量 workload、科研量 research、其他 other。

个体的定义采用了我们前面定义的 FOWL 语言, 举例如下:

```

<? xml version="1.0" encoding="UTF-8"?>
<temp>
  <fowl:Thing>
    <frdfs:name>武钰云</frdfs:name>
    <frdfs: workTime>3</frdfs: workTime>
    <frdfs:education>助教</frdfs:education>

```

<frdfs: workload>13</frdfs: workload>

<frdfs: research>27</frdfs: research>

<frdfs: other>3</frdfs: other>

</fowl: Thing>

</temp>

表明了个体武钰云目前的岗位是助教,在该岗位上工作了3年,教学量为13分,科研量为27分,其他为3分。

3) 个体隶属度的计算方法说明

本实验模拟的是教师评职称,前提是所有个体均为老师。评判指标如下:

对于每个个体,给其打分,由岗位年限(15分)、教学量(30分)、科研量(40分)、以及其他(15分)等4个因素组成,满分100分。其中:

岗位年限占15分,划为3个等级,工作0-3年得5分,3-5年得10分,5年以上得15分;

教学量占30分,根据在学校的教学任务等取值从0到30不等;

科研量占40分,包括发表论文、做科研项目等,取值从0到40不等;

其他占15分,属于额外加分或未考虑到的情况。

对于助教、教师、副教授、教授这些类,我们只关注了达到这些岗位所需要的分数。本系统中这样规定助教(60分)、讲师(70分)、副教授(80分)、教授(90分)。个体的隶属度就等于个体的总分与相应岗位职称分数之比。

每次演化时,我们采用的是随机改变个体的4个属性值之一,改变后再次计算个体对于某个岗位的隶属度,并展现出来。如果隶属度>100%,那么我们认为该个体已不再属于这个类,将其值重置为0。

6.2 系统的开发平台和工具介绍

6.2.1 开发平台

硬件: CPU: P4 2.4G, 内存: 256M DDR。

操作系统: Windows XP。

Web 服务器: Tomcat6.0。

6.2.2 开发工具

1) MyEclipse7.1

本系统选用 MyEclipse7.1 作为 Java 程序编辑器, MyEclipse 是基于 Java 的、开放源代码的可扩展开发平台,它只是一个框架和一组服务,用于通过插件组件构建开发环境。可以在同一集成环境中同时使用不同开发商开发的软件工具。其界面如图 6.4 所示:

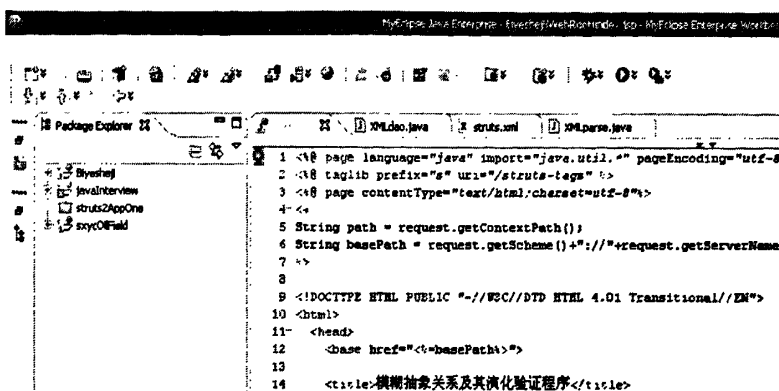


图 6.4 MyEclipse 界面

2) 数据库

采用 XML 文档作为数据库，存储方法如下：

```
<? xml version="1.0" encoding="UTF-8"?>
```

```
<temp>
```

```
<fowl: Thing>
```

```
<frdfs:name>武钰云</frdfs:name>
```

```
<frdfs: workTime>3</frdfs: workTime>
```

```
<frdfs:education>助教</frdfs:education>
```

```
<frdfs: workload>13</frdfs: workload>
```

```
<frdfs: research>27</frdfs: research>
```

```
<frdfs: other>3</frdfs: other>
```

```
</fowl:Thing>
```

```
</temp>
```

6.3 实验结果分析

职称晋升本体及其演化实验主要包括个体添加和演化两部分，分别实现个体的添加以及职称晋升本体的演化。试验系统的初始界面如图 6.5 所示：

界面中展现出来的信息有姓名、原始岗位（添加个体时的初始岗位）、岗位年限（占 15 分，不同年限分数不同）、教学量（累积教学量，0 至 30 分不等，包括上课课时及带研究生任务量等）、科研量（累积科研量，0 至 40 分不等，包括发表论文及科研项目等）、其他（占 15 分，我们没有考虑到，但需要额外加分的情况）、分值（个体的一个综合打分，即岗位年限、教学量、科研量、其他的总和）、助教隶属度（个体对于助教的隶属程度）、讲师隶属度（个体对于讲师的隶属程度）、副教授隶属度（个体对于副教授的隶属程度）和教授隶属度（个体对于教授的隶属程度）。、

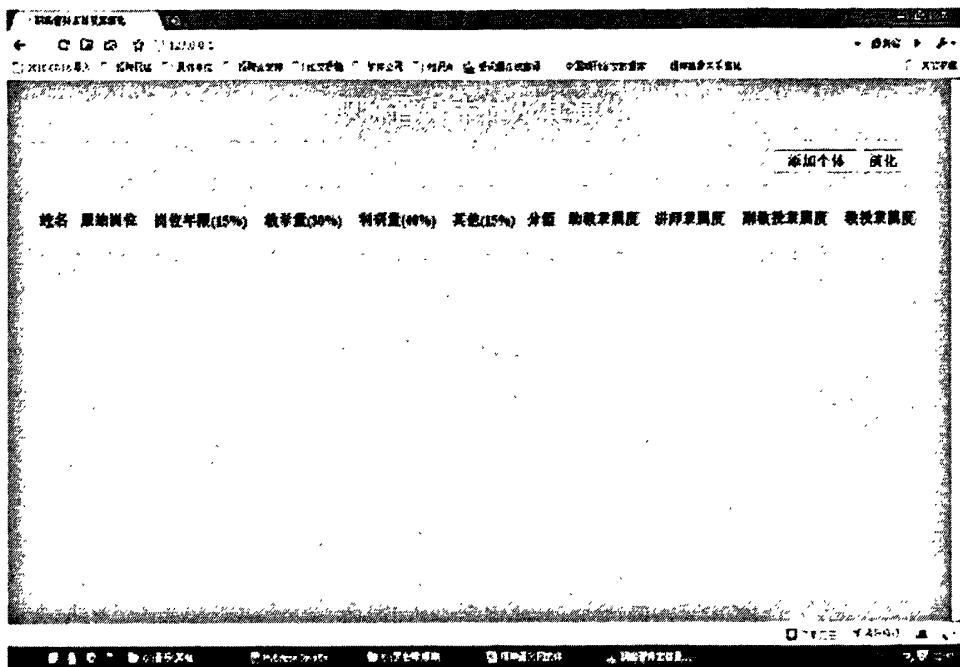


图 6.5 模糊抽象关系演化系统初始界面

1) 个体增加

点击“个体增加”按钮，增加一个个体，需要输入个体姓名、职称（包括助教、讲师、副教授、教授四个岗位）、岗位年限（可以选择 0-3 年、3-5 年、5 年以上）、教学量、科研量和其他（如图 6.6）。不同的选择对应不同的分数。

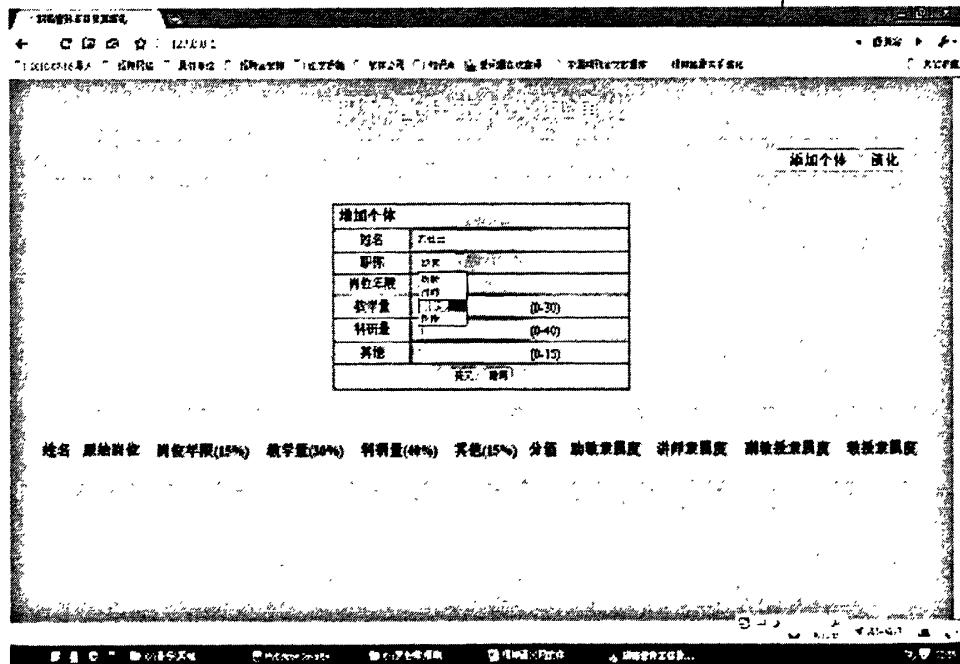


图 6.6 添加个体

添加个体 演化										
姓名	原始岗位	岗位年限(15%)	教学量(30%)	科研量(40%)	其他(15%)	分值	助教隶属度	讲师隶属度	副教授隶属度	教授隶属度
武柱云	助教	0-3年	1	1	1	8	100%	15%	2%	1%
王思衡	助教	0-3年	1	1	1	8	100%	15%	2%	1%
王娜	助教	0-3年	1	1	1	8	100%	15%	2%	1%
杨平	助教	0-3年	1	1	1	8	100%	15%	2%	1%
叶皓	助教	0-3年	1	1	1	8	100%	15%	2%	1%

图 6.7 演化前的个体

当信息录入完毕后，个体信息会展现在客户端，这其中包括了个体对于不同岗位的隶属度（如图 6.7）。我们可以看到，我们录入了五个个体，他们的初始数据都是一样的，都是助教且工作年限均为 0-3 年，并且教学量、科研量、其他均为 1。

2) 演化演示

添加个体 演化										
姓名	原始岗位	岗位年限(15%)	教学量(30%)	科研量(40%)	其他(15%)	分值	助教隶属度	讲师隶属度	副教授隶属度	教授隶属度
武柱云	助教	0-3年	4	1	1	11	100%	18%	2%	1%
王思衡	助教	0-3年	3	1	2	9	100%	15%	2%	1%
王娜	助教	0-3年	4	1	1	11	100%	19%	2%	1%
杨平	助教	1-3年	1	1	1	13	100%	21%	3%	1%
叶皓	助教	1-5年	1	1	1	13	100%	21%	3%	1%

图 6.8 一次演化后的个体

演化是通过随机更改个体的四个属性值之一,然后再次计算个体对于不同岗位的隶属度来计算的。图 6.8 是一次演化后的结果。

我们通过数据发现,5 个个体的初始值均为 8 分,对于助教的隶属度为 100%,因为他们的初始岗位即为助教。对于讲师的隶属度为 13%,对于副教授的隶属度为 2%而对于教授的隶属度仅为 1%。但是经过一次演化之后,我们逐个对比每个个体演化前后的数据:

表 6.1 个体演化前后对比表

	演化	岗位	教学	科研	其他	分值	助教	讲师	副教授	教授
武钰云	前	0-3 年	1	1	1	8	100%	13%	2%	1%
	后	0-3 年	4	1	1	11	100%	18%	2%	1%
王田甜	前	0-3 年	1	1	1	8	100%	13%	2%	1%
	后	0-3 年	1	1	2	9	100%	15%	2%	1%
王 寒	前	0-3 年	1	1	1	8	100%	13%	2%	1%
	后	0-3 年	4	1	1	11	100%	18%	2%	1%
杨 军	前	0-3 年	1	1	1	8	100%	13%	2%	1%
	后	3-5 年	1	1	1	13	100%	21%	3%	1%
叶 超	前	0-3 年	1	1	1	8	100%	13%	2%	1%
	后	3-5 年	1	1	1	13	100%	21%	3%	1%

通过以上数据我们发现,个体“武钰云”、“王寒”的教学量增加了,从原来的 1 变为了现在的 4,从而也影响到了他们对于讲师的隶属度从原来的 13%变为了现在的 18%;个体“王田甜”的其他分值由原来的 1 变为了现在的 2,从而也影响到了他们对于讲师的隶属度从原来的 13%变为了现在的 15%;个体“杨军”、“叶超”的岗位年限变化了,从原来的 0-3 年变为了现在的 3-5 年,从而也影响到了他们对于讲师的隶属度从原来的 13%变为了现在的 21%,对副教授的隶属度从原来的 2%变为 3%。

经过多次演化,演化结果如图 6.9 所示,我们可以发现,个体“王寒”、“王田甜”、“叶超”已经由原来的助教变成一个教师了,并有向副教授晋升的趋势,而这三个个体中,“叶超”对于副教授的的隶属度最大,很有可能最快晋升为副教授。而个体“武钰云”、“杨军”仍然停留在原来的助教岗位上,但是他们对于讲师的隶属度原来越大,在不久的将来,随着自己的努力,他们会成为讲师的。

300个教师个体演化

← 返回 保存 打印 退出

教师个体演化结果

姓名	原始岗位	岗位年限(15%)	教学量(30%)	科研量(40%)	其他(15%)	分值	助教教学质量	讲师教学质量	副教授教学质量	教授教学质量
沈任杰	助教	5年以上	19	5	11	50	100%	83%	12%	0%
王世刚	助教	5年以上	13	25	10	63	0	100%	90%	15%
王翠	助教	5年以上	19	17	11	62	0	100%	88%	15%
杨军	助教	5年以上	16	9	14	54	100%	90%	13%	0%
叶雄	助教	5年以上	16	25	9	65	0	100%	92%	10%

添加个体 演化

教师个体演化结果

图 6.9 多次演化后的个体

7 结 论

7.1 总结

本文从语义网的七层体系结构着手,重点研究了语义网中的核心层:RDFS、Ontology。结合了模糊认知图的理论,欲将 Web 上的信息资源赋予模糊性,这样 Web 上的资源间的联系及推理就像人对事物的认识一样,存在模糊性、演变性。换句话说,就是希望未来的网络能够像人一样的思考问题。本文在资源描述框架 RDF 的基础上提出了模糊资源描述框架的概念,意在描述网络上的模糊资源和资源间的模糊概念^[40]。同时给出了模糊本体的定义及表示以及模糊本体 Web 语言 FOWL。从而,就提出了模糊语义网的设想。并且在模糊语义网的基础上研究了本体层具体关系和抽象关系的演化。本文的主要工作包括以下几个方面:

1) 本文着重研究了资源描述框架 RDF。RDF 是一个网络资源对象和其间关系的数据模型,拥有简单的语义。采用形如“主体—谓词—客体”的三元组来描述 Web 上的各种资源和他们之间的关系,并提供一种基本的结构在 Web 上对这些元数据进行编码、交换和重用。RDF 以最低限度的约束,灵活的描述了网络上的资源。

2) 模糊认知图的理论。本文研究了模糊认知图的概念及表示模糊知识的形式。模糊认知图把知识存储在概念间的关系中,通过概念间的关系来表示模糊推理,概念节点的输出与两种类型水平有关,即概念节点自身的状态水平与外部因果联系的强度^[41],通过整个网络中各概念节点的相互作用来模拟系统行为,是一种无监督模型(unsupervised models)。

3) 模糊资源描述框架的提出。在深入研究了资源描述框架和模糊认知图的基础上,本文在 RDF 三元组中的谓词一项中加入了表示模糊度的权值。借助于模糊认知图的原理,来描述网络上资源间的模糊关系。

4) 模糊本体的定义及其形式化描述。在深入研究了本体的定义后,给出了模糊本体的定义,结合模糊资源描述框架 FRDF,描述了如何用 FRDF 表示模糊本体。将 RDF(S)层的模糊性传递至本体层,是对模糊语义网理论的进一步充实。

5) 模糊本体 Web 语言 FOWL。在研究了本体语言的基础上,提出了模糊本体 Web 语言 FOWL,给出了 FOWL 语法,并用其语法定义了属性、方法、个体、类。

6) 模糊语义网的演化研究。通过实验,研究 Web 上资源间的具体关系程度及所说的模糊度随外界因素及自身状态的变化而变化的情况。

7) 职称晋升本体及其演化实验。通过演化,不但个体的属性值变化了,而且随着个体属性值的变化,个体发生了质的改变,完全可能跳出了原来的类,属于不同的类。

通过对本课题的研究,了解了语义网的研究内容及进展。通过对本课题的研究,了解

到本体描述的是概念间的联系,用哲学的理论说“联系是永恒存在的”,所以哲学中的本体也是无穷大的。而本体语言即表示语言是语义网的核心,他提供了语义网中的领域信息描述的基础。具体实现领域的本体的设计是根据你所要实现的目标所建立的。所以本体在具体的研究领域中得到广泛的应用,但是要建立在各个领域中的本体是很困难的;对于本体描述语言,它也是随着人们的需求在不断的完善,完善的目的是既要不断的能够描述领域中复杂的关系,又要被计算机所理解;最后提出的模糊语义网的体系结构及对演化的研究,希望能为语义网的研究发展提供了一条新的思路。

7.2 展望

语义网构想了 Web 辉煌的未来。但在语义网的实现道路上还有很多尚待解决问题。本文在 RDF 的基础上提出了 FRDF 的思想,以表示 Web 上的模糊资源及资源间的模糊关系^[42]。并给出了模糊本体的定义及描述,同时定义了模糊本体 Web 语言 FOWL,这些工作充实了模糊语义网的理论,当然还有很多问题有待研究和解决。下一步的工作包括:

1) **FRDF 查询语言。**语义网的应用离不开查询技术的支持,针对不同的表示语言 XML、RDF 和 OWL,研究者们开发了众多的查询语言。在本文的后续工作中,需要支持 FRDF 的查询语言。

2) **模糊本体的自动生成技术。**本体的自动生成技术为本体库的建立带来了很多方便与好处,支持模糊本体的自动生成技术是未来语义网发展的需要。

3) **本体库的管理和维护。**本体间的模糊映射及本体内的概念间的模糊关系在 Web 漫长的发展与成长的过程中,其模糊性会发生演变。那么,就要求本体管理工具能够自动添加新产生的本体间的关系以及更新权值。

4) **概念间间接关系的获得方法与算法的研究。**概念间间接关系是指在具体关系描述层和抽象关系描述层中没有直接给出的关系,就要通过推理给出,这就需要给出推理规则和推理机制,当然也是模糊的。

5) **模糊本体管理系统(FOMS)原型系统的研究。**模糊本体的建立、本体元素的增删改、本体的解释与应用等。

参考文献

- [1] 陆建江,张亚非,苗壮,等.语义网原理与技术[M].北京:科学出版社,2007.
- [2] 杨晓青,陈家训.语义网[J].计算机应用研究,2002,23(6):42-45.
- [3] 刘柏嵩.基于知识的语义网:概念、技术及挑战[J].中国图书馆学报,2003,42(2): 32-35.
- [4] 张晓林.Semantic Web 与基于语义的网络信息检索[J].情报学报,2002,36(4):32-35.
- [5] 李卫华.语义网上本体与智能信息 Agent 的集成研究[J].计算机工程与应用,2002,7(16):111-114.
- [6] 黄伟 红,张福炎.基于 XML/RDF 的 MARC 元数据描述技术[J].情报学报,2000,(4): 57-60.
- [7] 任瑞娟,濮德敏,苗军民等.基于 XML/RDF 的 DC 元数据描述技术[J].情报杂志,2002, (9):71-74 .
- [8] 宓永迪,夏勇.资源描述框架(RDF)的应用[J].文献信息组织与利用,2001,(2):99-102.
- [9] 韩亚洪,许卓明.怎样在关系数据库中存储 RDF[J].计算机与现代化,2002, (11):12-15.
- [10] 袁勇智.一个基于 RDF/XML 自动更新的搜索引擎的设计及实现[J].业图书情报学刊,2003,(4):54-57.
- [11] 杨晓青,陈家训.一种利用 RDF(S)建立本体论的方法[J].计算机应用研究,2002,(4): 203-206.
- [12] 欧阳晔,姜浩,潘顺.XRQL:基于 XML 的新型 RDF 查询语言[J].微机发展,2003,(4):5-7.
- [13] 廖明宏.一个基于本体论的职工能力查询系统[J].计算机应用研究,1999, (12):33-36.
- [14] 陈文彬.Ontology 在图书服务网络中的应用[J].现代图书情报技术,2003,(6):6-9.
- [15] 邓凯,吴家春,王洪伟.本体论在知识图书馆中的应用初探[J].情报科学,2003, (3):32-36.
- [16] 楼向英.Ontology:概念及其在数字图书馆中的应用[J].图书馆杂志,2002, (11):25-29.
- [17] 李景,钱平,苏晓鹭.构建基于 Ontology 的知识门户[J].数字图书馆,2004,(2):107-110.
- [18] YuWei, Cao Jiaheng, Chen Junpeng. Novel method for searching ontologies on semantic web[J]. Journal of Southeast University(English Edition). Vol.22, No.3, Sept.2006.
- [19] Kang Dazhou, Xu Baowen, Lu Jianjiang, et al. Description logics for fuzzy ontologies on semantic web[J]. Journal of Southeast University(English Edition). Vol.22, No.3, Sept.2006.
- [20] 冯兰萍.本体在智能信息检索中应用研究[D]. 常州:河海大学,2005.
- [21] Axelrod R. Structure of decision: the cognitive maps of political elites[M]. Princeton, NJ: Princeton University Press, 1976.

- [22] Kosko B. Fuzzy cognitive maps[J]. Int J Man-machine Studies, 1986, 24.
- [23] Kosko B. Adaptive inference in fuzzy knowledge networks [C]. Proc 1st Int Conf Neural Networks, 1987, 2.
- [24] 张桂芸, 刘洋, 王元元. 基于模糊认知图的文本分类推理算法[J]. 计算机工程与应用, 2007, 43(12): 77-80.
- [25] MAGDALENA, KACPRZAK. Formalization of multi-agent reasoning [A]. Proceedings of the International Conference on Parallel Computing in Electrical Engineering [C]. Bialystok, Poland: [s. n.], 2002.
- [26] SUNR, FASLIM. Interrelations between the BD Iprimitives: Towards Heterogeneous agents [J]. Cognitive Systems Research, 2003, 4.
- [27] 虞和济, 陈长征, 张省, 等. 基于神经网络的智能诊断[M]. 北京: 冶金工业出版社, 2002. 6.
- [28] 史忠植, 王文杰. 人工智能[M]. 北京: 国防工业出版社, 2007. 2.
- [29] 虞和济, 陈长征, 张省, 等. 基于神经网络的智能诊断[M]. 北京: 冶金工业出版社, 2002. 6.
- [30] 魏海坤. 神经网络结构设计的原理与方法[M]. 北京: 国防工业出版社, 2005. 2.
- [31] 阳小华, 周龙骧. Web 用户的视图[J]. 软件学报, 1999, 5(10): 213-217.
- [32] 林鸿飞, 战学刚, 姚天顺. 基于概念扩充的中文文本过滤模型[J]. 计算机科学, 2000, 32(10): 12-15.
- [33] 邹景华. 语义网在智能检索中的应用研究[D]. 重庆: 重庆大学, 2005.
- [34] 张桂元, 贾燕枫. Eclipse 开发入门与项目实践[M]. 北京: 人民邮电出版社, 2006. 1.
- [35] Neches R, Fikes R E, Gruber T R, et al. Enabling Technology for Knowledge Shar-ing. AI Magazine, 1991, 12(56): 80-91.
- [36] Gruber T R. A Translation Approach to Portable Ontology Specifications. Knowledge Acquisition, 1993(5): 199.
- [37] Borst W N. Construction of Engineering Ontologies for Knowledge Sharing and Reuse. PPhD thesis. University Twente, Enschede, 1997: 67-72.
- [38] 孙吉红, 刘伟成. 基于语义网的信息过滤模型与算法[J]. 情报杂志, 2007(1): 2-4.
- [39] 邢春燕. 基于 XML_RDF 和 Ontology 的语义网之构建[J]. 沈阳教育学院学报, 2006(3): 128-130.
- [40] Perez A G, Benjamins V R. Overview of Knowledge Sharing and Reuse Components: Ontologies and Problem Solving Methods. In : Stockholm V R, Benjamins B, Chandrasekaran A, eds. Proceedings of the IJCAI299 workshops and Problem Solving Methods (KRR5) 1999: 1-15.
- [41] PATEL K, GUPTA G. Semantic Processing of the Semantic Web[M]. Heidelberg: Springer Berlin, Lecture Notes in Computer Science, 2003. 80-95.
- [42] T. Berners-Lee. The Semantic Web[J]. Scientific American, 2001(6): 1-6.

攻读硕士学位期间发表的论文

- [1] 罗钧旻,武钰云,吴彬.模糊语义网及其演化研究[J].西安工业大学学报增刊.2010,30:60-63.
- [2] 罗钧旻,武钰云,吴彬.模糊本体及其演化研究[J].微电子学与计算机.2011,28(5):140-143.

致 谢

三年的研究生生活转眼就过去了,时间过得真快,不得不感慨岁月不饶人啊!三年的研究生生活也许学到的好多东西以后工作并用不到,但作者以为,学到的这些理论会为以后发展奠定良好的基础。三年中,学到的更多的是如何去思考、如何认识自己,这才是最重要的。感谢西安工业大学提供给作者三年的学习机会。

毕业了不是结束,而是一个崭新美好的开始。在这里作者首先要感谢研究生指导老师罗钧旻教授,感谢他在这三年里给作者学习和生活上的帮助。尤其是在毕业论文课题上的帮助,从开题到中期再到预答辩,罗老师对工作认真负责,一次次深夜修改论文深深打动了作者。罗教授对工作的敬业让作者很佩服。在这里作者表示非常感谢。

其次感谢作者的师姐吴彬,感谢她在论文中做的工作,她提出的模糊资源描述框架FRDF(S),为作者的课题奠定了良好的基础,使作者的课题能继续进行。

再次感谢作者的师弟杨军,感谢他在实验程序上予以作者的帮助。

最后感谢在论文开题、中期答辩以及预答辩中给作者提出建议的各位老师。

还要感谢三年来陪伴作者度过的所有老师和同学们,感谢你们,作者的生活因为有了你们而精彩。

学位论文知识产权声明

本人完全了解西安工业大学有关保护知识产权的规定，即：研究生在校攻读学位期间学位论文工作的知识产权属于西安工业大学。本人保证毕业离校后，使用学位论文工作成果或用学位论文工作成果发表论文时署名单位仍然为西安工业大学。学校有权保留送（提）交的学位论文，并对学位论文进行二次文献加工供其他读者查阅和借阅；学校可以在网络上公布学位论文的全部或部分内容，可以采用影印、缩印或其他复制手段保存学位论文。（保密的学位论文在解密后应遵守此规定）

学位论文作者签名：

武钰云

指导教师签名：

郭俊

日期：

2011年4月22日

学位论文独创性声明

秉承学校严谨的学风与优良的科学道德,本人声明所呈交的学位论文是我个人在导师指导下进行的研究工作及取得的研究成果。尽我所知,除了文中特别加以标注和致谢的地方外,学位论文中不包含其他人已经发表或撰写过的成果,不包含本人已申请学位或他人已申请学位或其他用途使用过的成果。与我一同工作的同志对本研究所做的任何贡献均已 在论文中作了明确的说明并表示了致谢。

学位论文与资料若有不实之处,本人承担一切相关责任。

学位论文作者签名:

武钰云

指导教师签名:

王政

日期:

2011年4月22日