面向本体的MIS领域分析方法*)

房秀蓉　李师贤
(中国科学院计算技术研究所智能信息处理开放实验室　北京100080)
(中山大学计算机科学系　广州510275)

Ontology-oriented MIS Domain Analysis Method

FANG Xiu-Rong　LI Shi-Xian
(Computer Science Department, Zhongshan University, Guangzhou 510275)
susanf@public.guangzhou.gd.cn

Abstract Domain engineering is a reuse technology of large-grain size, it focuses the analysis, design and implementation within a specific domain. This paper focuses the first phase of domain engineering, discusses a new approach of domain analysis-ontology-oriented MIS domain analysis method and introduces the implementation of the prototype system which supports this method.

Keywords Software reuse, Domain engineering, Domain analysis, Domain modeling, Requirements analysis, Ontology

1. 引言

软件重用是解决软件危机的一个重要手段。实践证明，在特定领域中的重用可以获得更大的效益[3]。领域工程是一种大粒度的软件重用技术，它研究的是特定领域中的分析、设计和实现问题。领域分析是领域工程的第一个阶段，它的主要内容是确定领域边界、识别信息源、分析领域中的需求，最终建立领域模型，领域模型可以供后继的领域设计和领域实现使用[2]。领域分析阶段涉及的人员广泛：最终用户、领域专家、领域分析员等，和应用工程中的需求分析类似，采用一种标准的通信方式用于描述他们对领域的理解以获取统一的领域模型显得非常重要。

由于领域分析必须结合特定领域的专家知识，所以在一定程度上，可以认为领域分析是一种知识获取的过程。本文将讨论一种新的领域分析方法——面向本体的MIS领域分析(ONtology-Oriented MIS Domain Analysis)方法。ONODA方法在MIS领域中应用了一种新的知识表示方法——本体结合对象来描述领域分析阶段的结果：领域模型。本文还将介绍支持ONODA方法的原型系统，该系统支持用户已有的领域模型、领域知识库中的系统模型进行MIS领域中具体应用系统的需求分析。

2. 背景知识

2.1 领域分析

1981年，J.M. Neighbors在软件重用中引入领域分析的概念。他认为领域分析是“在特定问题领域的一组相似系统中识别对象和操作的行为”。领域分析与其它系统分析的最大区别在于领域分析关注的是一个特定领域中的所有相似或相近系统，而不是单一的系统。领域分析的最基本特征就是尝试识别一个领域中的多个系统间的共性和特性，领域分析阶段的主要目标是获取领域模型。

当前已经有许多研究领域分析的成果，著名的有：CMU/SEI于1990年提出的FODA(Feature-OrientedDomainAnalysis)方法[4]，IBM公司 Will Tracz 等人于1991年提出的 DSSA (Domain-Specific Software Architecture)方法[2]。上述两种方法对领域分析过程进行了完整的描述，把领域分析分为各个不同的阶段，确定各个阶段的任务。FODA 方法把领域分析分为三个阶段：上下文分析；领域建模；构架建模。DSSA 方法把领域分析分为5个阶段：定义领域范围；定义/提炼领域需求；定义/提炼领域设计和实现约束；生成领域模型/构架。但是，这两种方法只是提出了阶段性的指导意见，对于具体如何操作领域分析过程没有进行详细的描述。特别地，领域分析阶段中领域建模是最重要的部分，因此如何描述领域模型是一个关键的问题。

因此，设计一种形式化的语言描述领域专家的知识，既方便普通用户理解，又可以用计算机检查模型的完整性和一致性，是我们研究领域分析方法的一个目标。同时，我们希望能够利用计算机辅助用户进行领域分析，支持 ONODA 方法的原型系统就是作为一个 CASE 工具，自动完成领域分析过程中的一部分工作，减少领域分析过程中的出错率，从而达到缩短软件开发过程的目的，并为以后的领域设计和领域实现阶段做准备。

2.2 本体

本体原来是一个哲学概念。意大利本体学家 Guarino 认为本体是说明世界的某个方面的特定的分类体系，该体系不依赖于特定的语言。另一个本体学家 Roberto Poli 认为本体是关于对象和它们之间的联系的理论，本体提供了区别各种对象(具体或抽象的，存在或不存在的，现实或理想的，独立或依赖的)和联系的标准(关系，依赖，论断)。

在人工智能和知识工程界，本体最初是由 Neches 等人在1991年引入的。Mike Uschold 和 Michael Gruninger 等人认为本体指“在某个领域中共享的理解或是对共享概念的协定”，具体地，本体被表现为对一组词汇表的定义。最被广泛接受的本体定义是 Tom Gruber[5]于1995年给出的“本体是概念

*)本文得到广东省现代控制技术重点实验室和广东省教育厅软件技术重点实验室资助。

的显式表示”。具体地，本体是一组特定内容的知识表示原语的定义，包括类、关系、函数和对象常量等。

结合具体的领域，本体具有以下的特征：

·某个领域中的本体确认该领域的重要概念以及这些概念之间的关系，并进行精确的定义；

·这些重要的术语及其描述被所有参与该领域的人员所认同，这构成了领域内通讯的基础；

·本体可以独立于具体应用程序而设计，对于本领域中的其它应用程序来说是可以重用的。也就是：在应用程序之间可以用相同的领域本体来进行交流；

·一个本体可以被形式化，形式化后的本体可以支持 IT 系统间的交流。

本体作为知识表示的一种手段，由于它的标准性，因而具有以下作用：

·支持通信：包括人与人之间，人与系统之间，系统与系统之间的通信。

·互操作：由于本体的标准化，可以在使用了不同建模方法、语言、软件工具创建的系统之间进行互操作。

·在系统工程中，共享概念实际上表现了领域中的关键内容，将它们形式化之后就成了知识系统的可重用构件；并且系统的标准化表示使得系统的可靠性增加；再次，统一的共享概念有助于帮助确认系统的需求，定义软件系统的需求分析。

实验室进行的本体研究有斯坦福大学的 Ontolingua 项目；英国爱丁堡大学进行的企业本体(Enterprise Ontology)项目。随着基于 Web 的应用程序迅猛发展，本体的实验室研究成果逐渐走向商业应用。应用本体的著名企业有由 Tom Gruber 发起的 Intraspect (http://www.intraspect.com/)以及 Cycorp (http://www.cyc.com/) 和 Extricity (http://www.extricity.com/)等几家公司。它们主要销售基于本体的技术给用户，让它们与其它网站使用不同数据语义的用户可以进行“交流”。

3. ONODA 方法

3.1 核心思想

由于本体的标准化特征和描述概念、概念间联系的能力，在面向对象的领域分析方法基础上，本文讨论了一种新的领域分析方法——面向本体的领域分析(ONtology-Oriented Domain Analysis——ONODA)方法[1]，该方法关注的是管理信息系统领域。在 ONODA 方法中，形式化描述领域模型的语义网络(ONtology and Object-oriented NETwork——ONONET)和存放领域模型和系统模型的领域本体知识库(Domain Ontology Knowledge Base——DOKB)是核心。

ONODA 方法在建立领域模型时，用本体和对象结合的方式描述领域模型。我们知道，在传统的面向对象方法中，对象之间的关联通常采用 association、using 等关键字表示，而且，这些定义都是放在对象内部。然而在描述一个领域时，对象间的关系可能会错综复杂，把关系都封闭在对象中显然不利于领域专家和软件工程人员对领域知识的理解，所以我们考虑把关系从对象中独立出来。根据上文介绍，我们知道本体强调领域的本质概念，同时它也强调这些本质概念之间的关系。因此，在领域分析方法中引入本体概念，用以描述对象之间的关系。这样，综合应用本体和对象，可以更清晰地描述出领域模型，便于领域专家和最终用户理解领域模型。

3.2 领域建模

参考传统面向对象方法中的对象定义，我们定义本体和对象如下：

本体定义为一个七元组(N,F,P,A,B,R,M)，其中，N 是本体的名字，F 是本体的父本体名字，P 是指向本体涉及的和其它实体之间的关系指针，A 是本体的属性集合，B 表示组成本体的对象的集合，R 是本体网格，由 P 中的实体之间的关系组成，M 是附加到本体的方法集。

对象定义为一个五元组(N,F,P,A,M)，其中，N 是对象的名字，F 是对象的父类名字，P 是指向对象所涉及的本体指针，A 是对象的属性集合，M 是附加到对象的方法集。此处，对象的定义比传统的对象定义多了一个本体指针部分，用于描述对象和本体之间的关系。它和本体定义中关系指针对应。

ONONET(ONtology and Object oriented NETwork)语言是一种综合表示本体和对象的语义网络语言。用 ONONET 定义的领域模型是形式化的、可以被机器识别的，因此可以称之为一个程序，ONONET 程序的组成如图1所示。

```
〈ONONET 程序〉::=ONONET{〈程序名〉};
                    {〈介绍〉}
                    {〈一般描述〉}
                    {〈模块〉;}+
                    END_OF_ONONET
〈介绍〉::=Introduction
              Purpose:;
              Scope:;
              System_category:;
              end_of_Introduction
〈一般描述〉::=General_description
              General_function:;
              End_of_General_Introduction
〈模块〉::=〈对象模块〉|〈本体模块〉
〈对象模块〉::=Obj-Module(〈对象模块名〉);
              {〈对象树定义〉;}+
              end_of_module;
〈本体模块〉::=Onto-Module〈本体模块名〉;
              {〈本体树定义〉;}+
              end_of_module;
〈对象树〉::=Objclass(〈对象类节点〉):(〈对象节点序列〉)
            |Objclass〈对象类节点〉:(〈对象树序列〉)
〈本体树〉::=Ontoclass〈本体类节点〉:(〈本体节点序列〉)
            |Ontoclass(〈本体类节点〉:(〈本体树序列〉)
```

图1 ONONET 程序组成

根据本体和对象的定义，ONONET 语言总结定义了 MIS 领域中的13个对象类：

(1)岗位对象类；(2)数据表对象类；(3)数据库对象类；(4)数据仓库对象类；(5)视图对象类；(6)录入凭单对象类；(7)数据字典对象类；(8)模糊字典对象类；(9)数据变换格式对象类；(10)产生式规律对象类；(11)决策数规律对象类；(12)事件对象类；(13)态势对象类。

另外再定义了9个本体类，用于描述这些对象类之间的关系：

(1)组织型本体类；(2)机构型本体类；(3)岗位功能型本体类；(4)数据流型本体类；(5)数据演化型本体类；(6)复合视图型本体类；(7)复合态势型本体类；(8)规律发现型本体类；(9)体系结构型本体类。

这些对象类和本体类自身的纵向继承关系和它们之间的横向关系构成的网状结构组成了 MIS 领域的领域模型。以“机构型本体类”为例(如图2)，说明 ONONET 语言对本体和对象的定义规范，欲知细节请参考文[1]。

3.3 领域知识库

有了以上定义的领域模型，还需要建立一个存储领域模型以及后继生成的系统模型等领域知识的领域知识库(Do-

main Ontologies Knowledge Base——DOKB)。对这些领域知识的处理包括对它们的分类、组织和存储、浏览、查询、修改、生成。我们这样设计 DOKB:首先,采用树结构描述 DOKB 数据库的整体存储结构。利用 ini 文件的形式表示语义网络纵向层次关系中的一棵子树,其文件名对应该子树的根节点名,用 ini 文件中的段表示将子树分解后得到的一对多的基本数据单元。其次,进一步利用目录、子目录、索引文件和文件的层次结构表示更高的级别树形结构,具体如下:

知识库目录:dir = c:\ONONET

临时库目录:tempdir = c:\ONONET\Interkb

(1)用 ini 文件存储下层的子树,用 index.txt 索引文件存储上层的树形结构,其中的叶节点为下层子树对应的文件名;

(2)通过文件 dir\System\systemlist.txt 索引文件列出知识库中所有的系统;

(3)系统信息在 dir\System 目录下;

(4)每个应用系统的知识在以应用系统名为目录名的目录下;每个应用系统中有 index.txt 和 systeminfo.txt 两个索引文件,标识应用系统的对象树(包括超类)、本体树集合(包括超类)以及类、子类、实例的层次结构(不包括超类);

(5)应用系统中,每个类拥有一个目录;每个类的每个子类由在一个.cls 文件存储,每个类的每个实例放在一个文件中,文件名和实例名一致,扩展名则为 obj/ont;

(6)临时知识库设置的目的是为了暂时存放经过修改的知识库内容,经过语法和语义检查后才可以写入知识库;

```
Ontology(机构型本体类名)
        Father:name
        Attributes:
            Ontology Type=机构,
        aParameter(  岗位:(岗位对象名序列);
                     机构:(机构型本体类名序列);
                       岗位功能:(功能对象类名序列));
        Variables:
            Real Department:sequence of name
            Rea Position:sequence of (name,int)
        Ontonet:(机构设置网)
        Real Ontonet:(实际机构设置网)
        Methods:
            Create Class((子机构型本体类名),(机构设置网))
            Create Instance((机构型本体类名),(机构设置网))
            Print P-List
            Print R-List
        (Real Department)::=(机构型本体类名)
        (Real Position)::=((岗位对象类名),(人数))
        (岗位或部门)::=((岗位对象类名),(人数))
                       |(机构型本体类名)
                       |(岗位对象类名)
        (人数)::=(正整数)
        (机构设置网)::=(机构设置三元组序列)
        (实际机构设置网)::=(机构设置网)
        (机构设置三元组)::=
            ((岗位或部门序列);(基本关系名);(功能对象类名序列))
            |(岗位或部门序列);(基本关系名);(机构对象类名序列))
```

图2 ONONET 语言文本例子

3.4 支持 ONODA 的原型系统

支持 ONODA 方法的原型系统 Promis 4.0为用户进行领域分析和系统分析提供了一个计算机辅助工具。该辅助工具利用计算机完成一部分的分析工作,从而降低了分析工作的难度、减少了分析工作的出错率,使用户的分析工作能够更快速地完成。

(1)*系统功能*(图3)

·支持用户交互式进行的本体和对象设计

交互式是指提供向导性的界面。在这一部分,用户可以根据提示选择是需要继承已有的领域知识,还是由用户自己完全定义所需系统的需求模型;Promis 4.0系统将根据 ONONET 语言的语法、语义定义对用户的输入进行检查,如果发现用户输入有错误,则实时提醒用户,以保证对需求模型定义的一致性和完整性;用户必须对错误进行修改方可进入下一步的定义工作。定义好的系统模型将经过编译器的检查,最终写入 DOKB;

·支持用户交互式对本体和对象进行修改

这一部分的功能和上面的相类似,Promis 4.0系统提供交互式界面,指导用户对 DOKB 中已有的模型(领域模型和系统模型)进行修改。系统按照 ONONET 语法和语义对用户修改的内容进行检查,如果有错误,系统将提示用户;修改好的系统模型将再次通过编译器的检查,确认无误之后存入 DOKB;

·支持用户对本体和对象网状关系的查询

这一部分支持对已有的领域知识库的查询工作,ONODA 方法中定义的本体和对象的网状关系,将在这一部分以图形化的形式表现出来,充分体现了 ONODA 方法利用本体描述需求模型的优势。对于每一个需求模型,本体和对象之间的链接关系非常清晰,用户可以便捷地查询领域知识库中已经存储的各个需求模型,并可以选择符合自己要求的模块进行重用;为了用户的使用方便,在这一部分增加了修改子模块,功能和上一部分相同;

·支持用户用 DODL 语言定义整个系统的需求说明

这一部分提供了另一个定义需求说明的途径,较为高级的领域专家(即对 DODL 语言有较多了解)可以用 DODL 语言直接定义整个系统的各个模块,经过编译器编译之后,将 DODL 语言转换为 ONONET 语言形式,最终结果写入 DOKB。

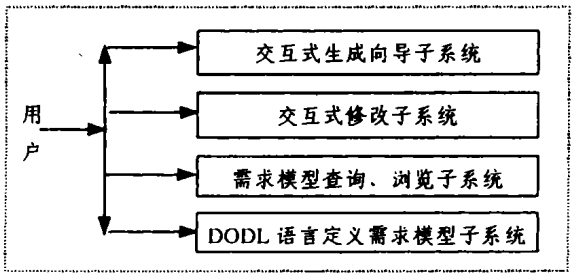

图3 系统结构图

(2)*接口设计* 在以上描述的四个子系统中,有三个接口很关键:知识库读写接口、DODL 语言编译器和 ONONET 语言编译器。知识库读写接口为交互式生成向导子系统、交互式修改子系统、需求模型查询、浏览子系统和 DODL 语言定义需求模型等子系统提供了和领域知识库 DOKB 交换数据的功能;ONONET 语言编译器在除了需求模型浏览、查询子系统之外的三个子系统中都提供了编译 ONONET 程序的功能;DODL 语言编译器则对用户输入的 DODL 程序进行编译。具体内容见文[8]。

(3)*实现环境* Promis 4.0系统的实现环境为:Windows NT 或 Windows 98操作系统,使用 Microsoft 公司 Visual C++6.0作为开发语言。

需要指出的是,使用 FLEX++和 BISON++程序生成 DODL 语言编译器和 ONONET 语言编译器。DODL 语言编译器生成过程如下:把 DODL 语言按照 LEX 语法写成后缀

(下转第182页)

名，JAVA 组件库通过获取存储在 UserStateObject 对象中的用户信息和.ini 系统配置文件中的信息，实时构建了这个列表，显示给用户，用户点击列表中的数据库名，就可以进入到该数据库的检索页面。图2所示的另外两个带红框的区域也是由 JAVA 组件库动态生成的。

值得说明的是，在数据库如此丰富的今天，每个学科都有相当多的数据库可供选择，一个数据库会有不同的用户检索系统提供服务，数据库用户界面的使用便利性会对用户的选择使用产生极大的影响。用户界面的使用便利性就是对于不同使用背景的用户，数据库都能够提供恰当的检索途径、丰富的检索策略编制工具、充分的在线帮助信息和灵活的检索结果的输出方法。SiteSearch 系统提供了一系列的功能帮助用户在系统所集成的各个数据库中查找信息，这些功能在 Web 页面上的体现全都是通过 JAVA 组件库完成。

图3显示了检索结果页面中由 JAVA 组件库生成的检索记录列表和检索记录浏览导航链接。中央的大块（带红框的）区域是检索记录列表，列出了馆藏记录中的名称和作者，类似的，生成这个区域的小器件 FormatRecordsWithDuplicates 类读取相关的记录格式显示配置文件，将记录格式化后显示给用户。通过修改格式显示配置文件，用户可以看到不同显示信息和显示效果。例如，可以把记录显示变成右对其，并为每条馆藏记录增加显示诸如出版单位、合作著者，著者单位等字段的信息。中央大块区域上方的横条（带蓝框的）是记录结果集的浏览导航链接，使用户方便地在各个记录之间进行浏览。

结束语 在数字图书馆的建设中，包括可定制界面在内的一系列用户个性化功能是其中一个十分重要的组成部分。SiteSearch 系统在以 Java 语言为主要软件开发平台的基础上，通过 API 中包含的 Java 组件库实现了用户界面的可定制功能。

基于众多因素的考虑，这个系统没有采用主流的商用开发模式，如 ASP、JSP 等动态网站制作技术，而是独辟蹊径，采用了 JaSSI 和 ZBase 组件与包含在 HTML 页面中的各种 Java 元素相结合、并辅以众多的系统配置文件的方式。这主要是因为，系统所连接的数据库都是针对图书馆界采用 Z39.50 协议标准的数据库，其协议的独特性在一定程度上决定了开发的独特性。

与“Web 服务器＋ASP/JSP＋数据库”的模式相比，SiteSearch 系统的开发模式较为复杂，需要 JavaPage 接口，实体和系统配置文件等多种机制的共同配合，但是它既能够满足针对 Z39.50数据库的跨库检索，又能实现统一的可定制用户界面的功能。而 ASP/JSP 作为较通用的技术，虽然也能完成动态用户界面，但只能访问常见的商用数据库，难以实现对图书馆界遵守 Z39.50协议的数据库的访问。

致谢 我们衷心感谢美国 OCLC 技术人员在该国际合作项目中给我们的大力支持和帮助。同时感谢该项目组的其他所有成员。

参 考 文 献

1 Oclc SiteSearch Online Help Documents. http://cypress.dev.oclc.org:7301/help/
2 Umlauft M. A Java Component User Interface for the Minstrel Push System. 2000
3 Peng Chengyuan, Vuorimaa P. Development of JAVA User Interface for Digital Television. Advanced Visual Interfaces, 2000. 276～279
4 Kwong Bor Ng. The applicability of universal pragmatics in information retrieval interaction: a pilot study, Information Processing and Management. Information Processing and Management, 2002, 38(2): 237～248
5 张德运，李德锴. 采用 JAVA 语言的通用型用户界面设计. 微型机与应用，1998
6 桂君，郭依群. 网络数据库用户界面的使用便利性研究. 情报理论与实践，2001
7 衡中青. Z39.50和 World Wide Web 的比较. 情报科学，2000

（上接第174页）

为.L 文件作为 FLEX++程序的输入，生成词法分析器；根据 DODL 语言的语法要求按照 YACC 语法写成后缀为.Y 文件作为 BISON++程序的输入，生成语法分析器。ONONET 语言编译器的生成过程和上述过程类似。关于 FLEX++程序、BISON++程序、.L 文件和.Y 文件的具体内容见文[8]。

结束语 领域工程是软件重用技术的一个重要分支和研究热点。领域工程的第一阶段——领域分析主要着重于解决领域建模问题。本文讨论了面向本体的 MIS 领域分析（ONODA）方法是指在一个特定的领域——管理信息领域中，定义一系列的领域本体类和对象类以构建领域的通用模型，为系统建模提供了前提条件；本文还讨论了支持该方法的原型系统的实现过程。在 ONODA 方法中，领域分析阶段被认为是一个知识获取的过程，因此，引入人工智能思想，采用本体作为知识表示方法，是 ONONDA 方法在理论上与一般领域分析方法的最大区别。虽然 ONODA 方法和 Promis 4.0系统在实现领域分析部分自动化工作中已经取得了一定的成绩，但是这部分工作还有需要继续完善的地方，如自动地获取领域知识、实现领域模型的修改。进一步地，我们希望 Promis 4.0 系统可以实现从领域分析阶段到后继的领域设计以及领域实现阶段的转换，将这三个阶段无缝地联接在一起，系统最终产生一个可运行的具体的管理信息系统。

参 考 文 献

1 Lu R, Jin Z. Domain Modeling-Based Software Engineering. Kluwer Academic Publishers, 2000
2 Sodhi J, Sodhi P. Software Reuse: Domain Analysis and Design Processes. McGraw-Hill Companies, Inc. 1998
3 Jacobson I, Griss M, Josson P. Software Reuse: Architecture Process and Organization for Business Success. 北京：世界图书出版公司北京公司，1998
4 Tracz W, Coglianese L, Young P. A Domain-Specific Software Architecture Engineering Process Outline. IBM Corporation Federal Systems Company
5 Gruber T. What is an Ontology? 1995
6 Uschold M, Gruninger M. ONTOLOGIES: Principles, Methods and Applications. Knowledge Engineering Review, 1996, 11(2)
7 Farquhar A, Fikes R, Rice J. The Ontolingua Server: a Tool for Collaborative Ontology Construction: [Technical Report KSL-96-26]. 1996
8 天鹰工作组：[天鹰项目技术报告]. 2000, 6