

面向本体的需求分析*

陆汝钫 金芝 陈刚

(中国科学院知识处理与人机交互实验室 北京 100080)

(中国科学院数学与系统科学研究院数学研究所 北京 100080)

(中国科学院管理、决策与信息系统开放实验室 北京 100080)

E-mail: {rqlu,zhijin,cg}@math03.math.ac.cn

摘要 提出了一种新的需求分析方法——面向本体的需求分析 OORA (ontology-oriented requirements analysis). 该方法的主要特点是,用本体作为对象关联的手段,以增强面向对象方法的表达能力. 文章首先定义了 OORA 的需求模型,然后结合实例讨论了信息系统需求分析中的两种主要的本体类型,最后给出了 OORA 方法的主要步骤.

关键词 需求分析,需求重用,本体,信息系统.

中图法分类号 TP311

面向对象分析在当前需求分析方法中最具代表性. 目前,研究者们已经提出了很多成功的方法,如 Jacobson 的 OOSE(object-oriented software engineering)方法^[1]、Rumbaugh 的 OMT(object modeling technology)方法^[2]、Shaler 和 Mellor 的 OOSA(object-oriented system analysis)方法^[3]以及 Coad 和 Yourdon 的 OOA(object-oriented analysis)方法^[4]等. 这些方法有一个共同点,就是它们都是基于单个对象分析的,正如文献^[5]所述,这些方法都是以对象和对象类为中心进行的,其他方面的属性,如对象属性、对象间的关系、对象间的消息传递、对象的用户界面等,则相对处于次要地位.

如果从一个更接近于我们所描述问题的角度来分析,不难发现如下基本事实:在一个大型软件工程项目的分析过程中,并不总是对象和对象类在起着首要的作用,有时对象或对象类之间的各种关系显得更为重要. 其主要原因是,首先,由于项目和软件本身可能极其庞大和复杂,要确切掌握不同对象或对象类之间的关系比理解单个的对象模块更为困难. 我们知道,构成物质的原子只有百余种,但由这些原子构成的分子的种类却无法计数. 软件工程师往往需要付出更多的时间和精力来分析对象类之间的关系,而不仅仅是只分析对象本身. 其次,从软件重用角度来看,软件开发者的注意力不应被个体对象的具体内容所吸引,而应集中考虑如何利用所获得的大量可重用软件构件去构造新的软件. 由此,软件构件之间的关系描述理应受到足够的重视. 最后,仅从软件构件的角度考虑软件重用还远远不够,软件框架的重用对大型软件开发来说更有价值,这实际上是大粒度软件重用的一种方式. 软件框架重用的一个重要方面是软件体系结构的重用. 目前已经有许多软件体系结构描述语言 (architecture description languages,简称 ADL)^[6],ADL 的两个主要部分就是构件以及描述构件之间关联的连接器.

综上所述,在需求分析过程中,除了需要识别对象或对象类、构造对象层次结构以外,另一个重要的方面就

* 本文研究得到国家自然科学基金重点基金(No. 69733020)、国家自然科学基金(No. 69983010)、国家“九五”重点科技攻关项目基金(No. 98-780-01-07-09)、国家 863 高科技项目基金(No. 863-306-ZT02-01-2)和国家教委留学回国人员科研启动基金(No. J9905)资助. 作者陆汝钫,1935 年生,研究员,博士生导师,中国科学院院士,主要研究领域为人工智能,软件工程. 金芝,女,1962 年生,博士,副研究员,主要研究领域为人工智能,软件工程. 陈刚,1968 年生,讲师,主要研究领域为基于知识的软件工程.

本文通讯联系人:陆汝钫,北京 100080,中国科学院数学研究所

本文 2000 01-17 收到原稿,2000-03-17 收到修改稿

是分析对象之间的其他关系以及对象的组合方式。事实上,几乎所有的面向对象分析方法都已经注意到这一点,它们中的大多数除了能表示对象的类层次结构外,还或多或少地提供了一些手段来表示对象之间的其他关联。如:许多面向对象方法都用 association, using, aggregation 等分别表示对象间的逻辑关联、引用关联和聚合关联等,这些关系名是相应方法中表示对象关联的词汇或语法。如果我们将这些词汇到软件设计的映射称为它们的语义,从这些词汇到应用描述的映射称为它们的语用的话,则针对当前的面向对象方法,人们可能会提出如下问题:(1) 这些词汇对表示对象关联是否足够,或对某类需求足够?(2) 它们的语义是否有明确的定义?(3) 它们的语用是否有明确的定义?提出第1个问题的必要性是明显的,足够的词汇保证足够的表达能力。第2个问题中所确定的语义可以规范从需求模型到软件设计的转换。而第3个问题的着眼点在于如何将应用领域描述准确地表达为需求模型。有效地解决这些问题,对于构造完整而准确的需求模型和进行有效的需求分析起着重要的作用。

本文提出一种新的需求分析方法——面向本体的需求分析 OORA (ontology-oriented requirements analysis),用以增强面向对象方法的表达能力,满足软件需求分析的要求。本体论最早是一个哲学上的概念,从哲学的意义上说,本体是指关于世界某个方面的一个特定的分类体系,这个体系不依赖于任何特定的语言。近年来,许多知识工程学者应用了这个概念。在知识工程界,最早给出本体概念的是 Neches 等人。在文献[7]中,他们将本体定义为“相关专题的基本术语和关系,以及利用这些术语和关系构成该专题的规则的组合”。后来,Gruber 在 1998 年又给出了本体的一个更为流行的定义,即“本体是领域概念模型的显式表示”^[8]。具体来说,某个领域的本体就是关于该领域的一个公认的概念集,其中的概念含有公认的语义,这些语义通过概念之间的各种关联来体现。本体通过它的概念集及其所处的上下文来刻画概念的内涵。总之,本体强调相关领域的本质概念,同时也强调这些本质概念之间的关联。我们提出的 OORA 方法就是借鉴本体论研究的这种观点,在面向对象方法中引入对象关联,并给出各种关联的语义语用内涵,试图建立一种新型的需求模型,支持需求获取和需求分析的自动化。

1 对象和本体的结合

观点 1. 对象是本体的基本元素

概念是本体的基本组成元素。在目前许多关于本体的研究中,本体中的概念是无结构的,这些概念不能再分为更细的组成成分。我们在仔细研究了信息系统设计中本体的应用后发现,如果我们不考虑这些概念元素的结构,则本体似乎并不具备足够的描述能力和模型构造能力。但是,如果我们假定本体是一组结构化成分的组合,那么这些成分又将是什么呢?我们对该问题的回答自然而又直接,即它们都具备对象的结构。换句话说,我们把对象视为本体的基本组成成分。这里的对象不仅起到本体内部元素的作用,在信息系统的分析和领域建模中,对象和本体具有同等重要的意义,就像点和弧在图中所起的同等作用一样。

观点 2. 用关系连接对象

在面向对象方法中,对象和对象类构成层次结构。它们按一种垂直方式组织,这种垂直的组织方式仅说明元素之间具有父子关系。在现实世界中,对象之间还存在着其他重要的关系。例如,考虑 3 个对象:人、计算机和文稿,三者之间并不存在任何父子关系,而“人使用计算机撰写文稿”这一情景却隐含着如下关系:人是使用计算机的用户,计算机是撰写文稿的工具,人是文稿的作者。如果把把这些关系简单地看作是某一对象的属性显然不太自然。一种较好的办法是用“文稿撰写”之类的动作关系作为本体将所有这 3 类对象组合起来。因此,关系在组成本体的各对象之间起到了至关重要的作用。

用本体连接对象还有另外一个优点。我们曾经谈到,增大对象的粒度是面向对象方法中用来降低软件复杂性的基本技术。对象聚类是增大对象粒度的基本方法。但是,由于简单的对象归并操作没有对这些对象的功能关系和界面作出严格定义,常常导致对象状态的混乱。而用本体连接对象则有可能解决这个问题。这是我们提出用本体连接对象的另一个理由。

观点 3. 本体之间存在继承层次

由前述两个观点我们可以得到 OORA 中的本体的一个直观认识,即本体是一个由对象节点和关系节点以

及这些节点之间的连线组成的网络。那么在我们的需求分析过程中,是只有一个全局本体,还是存在多个本体但它们同处于一个抽象层次,或存在多个本体且它们按某种抽象原则形成继承层次?我们的回答是第3种情况。不同层次的对象组成的本体形成不同抽象程度的本体,并形成本体间的继承层次,这种结构除了具有模块化、封装性等好处外,对需求分析的最大好处在于,父本体作为对子本体的一种约束,成为需求一致性和完整性检查的主要约束条件。

2 OORA 需求模型的结构

2.1 对象作为基本元素

任何信息系统都具有下面3种基本成分:执行活动的角色(如岗位、机构、部门,或其他同类名词)、角色执行的活动和活动作用的数据。因此,我们定义3种主要的对象。

定义1. 定义如下3种基本对象类型:

(1) 角色对象。一个角色是一个社会机构、一个岗位或一个计算机程序,角色对象描述角色可能执行的活动,并指定角色访问特定数据时的权限,其表示范式为

角色对象 \rightarrow 多个活动+多个(数据,访问权限)对。

(2) 活动对象。一个活动是一个数据处理过程,或一个抽象的动作,活动对象定义输入数据、输出数据和基于输入数据产生输出数据的处理操作,其表示范式为

活动对象 \rightarrow 多个输入数据+多个输出数据+单个(多参数)操作。

(3) 数据对象。一个数据对象是一个结构型数据,它用一组属性来规范这些数据,并定义了在这些数据上可能执行的动作的集合。我们把可以修改的数据对象称为数据池对象,数据库和数据表就属于数据池对象,不能修改的数据对象称为数据工具对象,数据字典和数据仓库则属于数据工具对象。数据对象的表示范式为

数据对象 \rightarrow 数据结构(带数据)+多个(单参数)活动。

2.2 建立对象间的关联

定义2. 对象之间可能存在如下7种关系:

- (1) 存在于多个角色之间的关系;
- (2) 存在于多个数据之间的关系;
- (3) 存在于多个活动之间的关系;
- (4) 存在于多个角色和多个活动之间的关系;
- (5) 存在于多个角色和多个数据之间的关系;
- (6) 存在于多个数据和多个活动之间的关系;
- (7) 存在于多个角色、活动和数据之间的关系。

综合这些关系和信息系统需求的结构,我们提出如下信息系统需求本体中的主要关系。

定义3. 构成信息系统需求本体的主要关系有:

机构关系:Org(角色集,社会关系,角色集);

数据传递关系:Trau(角色,数据集,角色集);

活动分解关系:Sub-Act(活动,活动集);

数据处理关系:DP(数据集,活动,数据集);

角色处理关系:RP(数据集,角色,活动,数据集);

工具处理关系:TP(数据集,数据工具集,活动,数据集);

混合处理关系:MP(数据集,角色,数据工具集,活动,数据集)。

这里,角色集是角色对象的集合,角色指角色对象。数据集是数据对象或工作缓冲区的集合,社会关系是一种标志,它的含义依赖于上下文,并在相应的应用程序环境中定义。活动指活动对象,数据工具集是一组数据工具对象的集合,数据工具指数据工具对象。其中DP,RP,TP和MP都具有数据处理功能。

例如,Org({主管,部门主管},领导,{部门})是一个机构关系.MP({报表1,报表2},数据分析师,{数据字典,模糊数据字典},月记帐,{月总结报告})是一个混合处理关系.

定义 4. 本体定义为一个六元组(N,F,A,B,R,S),其中 N 是本体的名字;F 是 N 的父本体名;A 是本体的属性集合;B 是组成该本体的对象集合;R 是本体网格,由 B 中元素之间的关系组成;S 是附加到该本体的方法集.

定义 5. 设 r 是一个本体关系,则 op(r)代表 r 的关系符,arity(r)表示 r 的参数个数,arity/n(r)表示 r 的第 n 个参数.

定义 6. 设 b1,b2 是两个对象,则 b1<b2(或 b2>b1)表示 b1 是 b2 的祖先. b1≤b2(或 b2≤b1)表示 b1=b2 或 b1<b2.

在上述六元组的 6 个元素之中,最重要的是 B 和 R. 因此,如果不致引起二义性,我们可以只用三元组(N,B,R)来表示一个本体.

2.3 用网络同态概念来定义本体的继承关系

定义 7. 令 O1=(N1,B1,R1)和 O2=(N2,B2,R2)是两个本体. 我们说存在一个由 O1 到 O2 的同态映射 M, 如果:

- (1) 映射 M 由 3 个局部映射组成: M(O1)=M(N1)∪M(B1)∪M(R1),
- (2) M(N1)=N2,
- (3) M(B1)∪{M(e1)|e1∈B1}, 其中,对 B1 中的每一个元素 e1,有且仅有 B2 中的一个元素 e2,满足 M(e1)=e2≥e1,
- (4) M(R1)=∪{M(r1)|r1∈R1}, 其中,对每个关系 r1∈R1,存在关系 r2∈R2,满足:
 - (a) op(r1)=op(r2),
 - (b) arity(r1)=arity(r2),
 - (c) 对 i=1 到 arity(r1),如果 arity/i(r1)是一个集合,则对arity/i(r1)中的每一个元素 e1,有且仅有 arity/i(r2)中的一个元素 e2,满足 M(e1)=e2≥e1,否则,令 e3=arity/i(r1),e4=arity/i(r2),则有 M(e3)=e4≥e3.

命题 1. 令 O1,O2,O3 是 3 个本体,如果存在一个从 O1 到 O2 的同态映射 M1 和从 O2 到 O3 的同态映射 M2, 那么组合 M2(M1)是从 O1 到 O3 的一个同态映射.

证明:结论明显成立.

定义 8. 令 O1=(N1,F1,A1,B1,R1,S1)和 O2=(N2,F2,A2,B2,R2,S2)是两个本体,我们说 O1 是 O2 的父本体(或 O2 是 O1 的正则子本体),如果:

- (1) F2=N1,
- (2) A1 是 A2 的子集,
- (3) S1 是 S2 的子集,并且
- (4) 存在一个从(N1,B1,R1)到(N2,B2,R2)的同态映射.

当然,在某些特定环境中,子本体并不需要完全继承父本体内的所有属性. 这意味着,子本体可能与父本体在同一属性上的取值不同;同样,子本体与父本体在执行方法上也可以有所不同. 这些特征超出本文讨论的范围.

3 静态本体和动态本体

下面给出两个典型的 OORA 分析实例.

首先,让我们考虑一个医院管理信息系统的例子. 该系统包含一个数据资源集,它由医院人事数据库、病人库、传染病人数数据库、药品库所组成. 为医学研究方便起见,该系统采用数据过滤和数据采掘等技术,从原始数据库中获取大量历史数据建立了一个数据仓库. 例如,数据仓库记录了所有失败的外科手术案例和所有复杂但

仍然成功的治疗案例等。该系统还能提供大量的数据表和视图数据,供医生、护士和管理人员使用,例如,记录目前需要特别护理的病人数据表、记录重要病人和严重病人病情转变情况表等。该系统还可提供一些数据表和视图供院外用户使用。例如,提供给医药卫生管理部门的传染病案例和传染源分布的数据表。上述的数据表和视图采用各种不同的表现形式,如图表和柱状图。该系统通过多种不同的用户界面将数据输入到数据源中,包括图形方式输入、数据扫描输入和交互式输入等。还有大量的小应用程序(applets)用于特定问题的处理,如专门用于数字统计的计算程序、数据采掘程序、例外处理程序等。

将上述信息组织成 OORA 中的一个本体,则得到如下结构:

对象:

- | | |
|------------------|---------------------|
| 源数据元素(data): | (1) 医院人事库, |
| | (2) 病人库, |
| | (3) 药品库, |
| | (4) 传染病人库, |
| | (5) 外科手术库, |
| 生成和保存数据元素(data): | (6) 失败外科手术案例数据仓库, |
| | (7) 成功治愈复杂疾病案例数据仓库, |
| 现场数据元素(data): | (8) 特别护理病人表, |
| | (9) 重要病人或病情严重病人表, |
| | (10) 当前传染病例及其区域分布表, |
| 数据分析元素(role): | (11) 数据统计包, |
| | (12) 数据仓库生成器, |
| | (13) 数据挖掘程序, |
| | (14) 实时例外处理程序, |
| 外部支持元素(tool): | (15) 数据字典, |
| | (16) 模糊概念词典, |
| 外部通道(role): | (17) 数据输入界面, |
| | (18) 图形输入界面, |
| | (19) 数据表输出, |
| | (20) 图形输出, |
| 活动(activity): | (21) 数据录入, |
| | (22) 数据查询, |
| | (23) 案例生成。 |

关系:

- (a) 输入: $RP(\{(17), (21)\}, \{(1), (2), (3), (4), (5)\})$,
- (b) 产生: $RP(\{(2), (5)\}, \{(12), (23)\}, \{(6)\})$,
- (c) 产生: $RP(\{(2), (3), (5)\}, \{(12), (23)\}, \{(7)\})$,
- (d) 输出: $DP(\{(2), (5)\}, \{(8), (9)\})$,
- (e) 输出: $DP(\{(3), (4)\}, \{(10)\})$ 。

以上是医院管理信息系统中的一个本体的简化描述,许多细节诸如元素的属性、关系的属性、数据库的操作和系统的用户等均被忽略。

现在让我们考虑另一种类型的本体:医院医疗设备购买本体。这个从申请购买医疗设备到该设备到达,并在医院的财产清单上登记注册的过程,在 OORA 中可以表示为如下本体:

对象:

- | | |
|-----------|-----------|
| 角色(role): | (1) 申请者, |
| | (2) 医院领导, |
| | (3) 会计师, |
| | (4) 投标者, |
| | (5) 评价者, |

| | |
|---------------|-----------------|
| 现场数据元素(data): | (6) 购买设备的预算报告, |
| 源数据元素(data): | (7) 医院的现金帐务, |
| | (8) 医院本年度预算, |
| 数据分析元素(role): | (9) 配置优化器, |
| 支持型元素(data): | (10) 设备生产者企业清单, |
| | (11) 医疗设备价格清单, |
| 输入数据(data): | (12) 申请报告, |
| | (13) 签字批复, |
| 收到数据(data): | (14) 销售者标书, |
| 输出数据(data): | (15) 医疗设备, |
| 发送数据(data): | (16) 宣布中标者名单, |
| 活动(activity): | (17) 数据录入, |
| | (18) 设备优化. |

workflow:

- (a) 输入: $RP(\{\}, (1), (17), \{(12)\})$.
- (b) 传输: $Tran((1), \{(12)\}, \{(9)\})$.
- (c) 产生: $MP(\{(12)\}, (9), \{(11)\}, (18), \{(6)\})$.
- (d) 传输: $Tran((9), \{(6)\}, \{(1)\})$.
- (e) 传输: $Tran((1), \{(6), (7)\}, \{(2)\})$.
- (f) 总结: $RP(\{(6), (7)\}, (2), (17), \{(13)\})$.

上述第 1 个例子描述了数据源之间的数据流关系,第 2 个例子描述了数据处理过程中的 workflow 关系,这些都是 OORA 方法所关注的主要本体类型.我们称第 1 种本体为静态本体(static ontology),称第 2 种本体为动态本体(dynamic ontology).它们共同构成 OORA 本体簇的基础.

4 OORA 方法

4.1 面向最终用户和自动开发的策略

一般地,软件需求分析由软件工程师来完成,但是,他们通常都不是领域专家,因此,要让他们理解应用领域的全部细节比较困难.另一方面,领域专家也不熟悉软件工程的原理,他们提供的需求定义常常不完全,并且经常要作变更.由于缺少共同语言,这两类人之间的合作都不很成功.结果是导致基于这种需求的软件不能付诸实践.

如果我们能开发出一种让软件工程师和最终用户都可以接受的语言,那么它将在消除这两类人之间的通信障碍方面起到重要作用.更进一步地,如果我们能找到一种可以使这种语言可执行的方法,比如,有一种方法如果能够把用这种语言写出来的任何代码转换成一个可执行的信息系统,那么应用开发的过程几乎可以自动化.这种自动方法实现的关键是开发出一个领域知识库,它支持把上面提到的定义转换成一个可以运行的信息系统.OORA 就是基于这个观点提出来的,它不仅面向软件工程师,而且还面向最终用户.

4.2 OORA 方法的主要步骤

基于上述总体策略,OORA 分 4 个阶段完成.

第 1 阶段.用一种类自然语言 BIDL(business information description language)描述企业事务.

第 2 阶段.确认隐含在 BIDL 文本中的本体和对象.

第 3 阶段.将这些本体和对象转换成另一种语言 Ononet(ontology and object-oriented network),得到用 Ononet 书写的需求预定义.

第 4 阶段.在采用 Ononet 作为知识表示形式的领域本体知识库中搜索相关的知识,并和前面的需求预定义合并,得到完整的需求定义.

注意,完整性和一致性检查在 OORA 中没有形成一个分立的阶段,这是因为:

- (1) 在开发/维护/演化阶段,领域本体知识库的一致性和完整性已经得到了保证;
- (2) 合并知识的完整性和一致性检查包含在合并过程中。

4.3 BIDL 语言

BIDL 是一种类自然语言.先解释一下类自然语言的概念,这种语言有 3 个主要特征。(1) 它非常接近于自然语言,因此可以将它看作是中文自然语言的一个子集;(2) 它不包含软件工程和软件技术的术语,所以领域专家们可以很容易地用它来描述他们的应用知识;(3) 它是一种可被编译器分析和理解的形式语言。

以下是 BIDL 语言的一个简要介绍,每个 BIDL 程序描述一个具体应用的情况,它被组织成如下 7 段:

- (1) 企业的总体特性,
- (2) 企业要处理的任务,
- (3) 企业内外部的职权范围,
- (4) 和企业相关的外部信息,
- (5) 企业的信息结构,
- (6) 企业的决策策略,
- (7) 企业的背景和统计数据。

下面,我们以“地方税务局”信息管理系统为例来示例 BIDL 的特征:

- (1) 地方税务局由市政府领导,
- (2) 地方税务局的**任务**包括所得税管理、消费税管理、运输执照税管理。
- (3) 地方税务局的**领导**负责整个地区的地方税务管理。
- (4) 对每个税种,要维护一个全地区的收税数据库。
- (5) 运输执照税数据库包括 3 个表:年度小汽车执照税务表、年度卡车执照税务表、年度自行车执照税务表。
- (6) 每月,打印数据表给没有及时交税的公司,打印数据表给各交税的公司。
- (7) 每迟交税 1 个月增加滞纳金罚款百分之一。
- (8) 整个地区有 137 家高科技公司,他们可以免税 30%。

BIDL 语句的语法规定了一套关键字(上面给出的例子中用黑体和仿宋体表示)。关键字分为两级:系统级关键字和领域级关键字。前者用常规的黑体打印,而后者用仿宋体字。

有关 BIDL 语言的更详细的描述参见文献[9]。

4.4 本体和对象的识别

需求的原始信息可以通过与专家对话或从现有文字材料中获取。修改作为需求分析起点的 BIDL 文本就可以较容易地重写原始信息,利用类自然语言极大地简化了我们的工作。

对 BIDL 文本的面向本体分析可以用自顶向下或自底向上或两者混用的方法。自顶向下分析首先标识出本体集,再找出它们的构件——对象;自底向上分析首先标识的是对象,再根据本体的概念将它们分类;当两种方法混用时,确定对象(类)和本体是交互进行的。

算法 1. 以“活动”为中心的自顶向下方法。

- (1) 标识出领域最顶层的本体概念,即顶级本体;
- (2) 标识出领域中所有的活动并构成一个集合,记为 Act ;
- (3) 标识出 Act 集合中的所有父-子关系,并形成 Act 中的继承层次;
- (4) 如果某个活动 x 可以被分解成子活动,那么认为 x 是一个本体并且把它的所有子活动加入到 Act 中;
- (5) 如果 x 不能被分解成子活动,那么认为 x 是一个活动对象;
- (6) 对 Act 中的每一个活动,确定活动的作用者;对 Act 中的每一个活动对象,确定活动操作的数据。

例:

- (1) 语句“ x 由 y 领导”蕴含:存在一个表示机构关系的静态本体, x 和 y 都属于这个本体;
- (2) 语句“ x 的任务包括 y, z, \dots ”蕴含:存在一个数据处理类型的动态本体 x ,同时蕴含:在 y, z, \dots 和 x 之

间存在着部分-整体关系;

(3) 语句“ x 被分为 y, z, \dots ”蕴含: 在 x 和 y, z, \dots 之间存在着父-子关系;

(4) 如果实体 y 出现在语句“ x 的任务包括 y, z, \dots ”中, 但是不存在语句“ y 的任务包括 u, v, \dots ”, 蕴含: y 是一个活动对象。

算法 2. 以“角色”为中心的自底向上方法。

(1) 标识出所有文本中提及的角色, 构造一个 rol 集;

(2) 对 rol 集中的每一个角色 x , 找出所有由这个角色操作的活动, 构造集合 $Act(x)$;

(3) act 是 $Act(x)$ 中的一个活动, 如果 act 不可分解, 那么认为它是一个活动对象, 否则是一个本体, 将其所有子活动加入 $Act(x)$ 中;

(4) 如果 $Act(x)$ 中的几个活动可以组成一个更大的活动, 那么为这些活动创建一个本体, 加入 $Act(x)$ 中;

(5) 合并所有的活动集构成一个集合, 记为 Act ;

(6) 标识出 Act 集合中的所有父-子关系, 并形成 Act 的继承层次;

(7) 对 Act 中的每一个活动对象, 确定活动操作的数据。

4.5 Ononet 语言

Ononet 语言是一种综合表示本体和对象的形式化语言。在这种语言中, 对本体和对象分别给出了明确的定义。对象定义与面向对象语言中的对象定义基本一致, 只是增加了指向它从属的本体的指针, 这些指针在参数 (parameter) 部分以如下形式给出定义:

$$\text{Parameter-of}(r_1: on_1, r_2: on_2, \dots, r_n: on_n)$$

这里, on_i 是对象从属的本体的名称, r_i 是对象在相应本体中的作用。

本体定义在名称、父类、属性、方法上与对象定义基本一致, 但有两个主要的区别。(1) 本体只包含指向它所联系的对象指针, 表示为如下形式:

$$\text{Parameter-of}(r_1: ob_1, r_2: ob_2, \dots, r_n: ob_n)$$

这里, ob_i 是本体联系的对象名称, r_i 是对象在这个本体中的作用。(2) Ononet 是一个本体网络, 它描述了对象如何通过本体互相联系。Ononet 中关系的表示已在第 2 节给出, 它的工作机制也已在第 3 节给出详细的讨论。

5 结 论

我们在 90 年代初就开始着手做 PROMIS (prototyping management information systems) 项目^[9~12], 其主要目的是在领域分析和领域模拟的基础上自动生成信息系统。在 PROMIS 中, 我们主要依靠两种类型的知识, 即软件工程知识和领域知识。除这两类知识外, 类自然语言 BIDL 是该项目的一个主要特点, 该语言不包含软件工程方面的专业术语, 并且十分接近于自然语言表达方式, 任何一个没有计算机编程经验的领域专家都可以用它来描述有关应用的具体信息。另一方面, 用这种语言编写的程序 (实际上是一个文本) 可以像通常的计算机程序一样被理解和编译, 整个过程得到上述两类知识的支持, 编译的结果是一个可在 Client/Server 环境下直接运行的信息系统。基于以上理由, 我们称支持上述整个过程的方法为 KISSME 方法 (knowledge intensive software system manufacture engineering)^[13]。

本文进一步提出结合面向对象和面向本体的技术。综合分析现有的需求分析方法, 可以发现两种基本的模式: 以控制流为中心的需求分析和以数据流为中心的需求分析。前者的典型代表是 Jackson 方法, 而面向对象方法恰恰是后者的代表。这两种方法到底孰优孰劣还没有定论, 本文提出的 OORA 方法试图将上述两种方法结合到一个模式中, 希望能综合利用这两种方法的优点。

致谢 我们感谢杨美清教授和她的青岛软件工程组对 PROMIS 项目的有力支持。我们也感谢李师贤教授和他的工作小组与我们的成功合作。

参考文献

- 1 Jacobson I, Christerson M, Jonsson P *et al.* Object Oriented Software Engineering: A Use Case Driven Approach. New York: Addison-Wesley Publishing Company, 1992
- 2 Rumbaugh J, Blaha M, Premerlani W *et al.* Object-Oriented Modeling and Design. Englewood Cliffs, NJ: Prentice Hall, Inc., 1991
- 3 Shlaer S, Mellor S J. Object-Oriented Systems Analysis: Modeling the World in Data. Englewood Cliffs, NJ: Yourdon Press/Prentice Hall, Inc., 1988
- 4 Coad P, Yourdon E. Object-Oriented Analysis. 2nd edition, Englewood Cliffs, NJ: Prentice Hall, Inc., 1991
- 5 Shao Wei-zhong, Yang Fu-qing. Object-Oriented Systems Analysis. Beijing: Tsinghua University Press/Guilin: Guangxi Publishers of Science and Technology, 1998
(邵维忠, 杨芙清. 面向对象的系统分析. 北京: 清华大学出版社/桂林: 广西科学技术出版社, 1998)
- 6 Shaw M, Garlan D. Software Architecture. Englewood Cliffs, NJ: Prentice Hall, Inc., 1996
- 7 Neches R, Fikes R E, Finin T *et al.* Enabling technology for knowledge sharing. AI Magazine, 1991, 12(3): 36~56
- 8 Gruber T R. A translation approach to portable ontology specifications. Knowledge Acquisition, 1993, 5(3): 199~220
- 9 Lu Ru-qian, Jin Zhi, Wan R. Requirement specification in pseudo language in PROMIS. In: Chang S *et al.* eds. Proceedings of COMPSAC'95. NY: IEEE Computer Society Press, 1995. 96~101
- 10 Lu Ru-qian, Jin Zhi, Wang Rong-lin *et al.* An approach of acquiring requirement information based on domain knowledge. Journal of Software, 1996, 7(3): 137~144
(陆汝钫, 金芝, 万荣林等. 基于领域知识的需求信息获取. 软件学报, 1996, 7(3): 137~144)
- 11 Lu Ru-qian, Jin Zhi, Xia Y. An intelligent case tool for developing information systems. In: Ju D *et al.* eds. Proceedings of the International Symposium on Future Software Technology'96 (ISFST'96). Xi'an: Xi'an Communication University Press, 1996
- 12 Lu Ru-qian, Jin Zhi, Liu L *et al.* PROMIS V2.0: an intelligent CASE tool for MIS in the client/server application. In: Zhong X *et al.* eds. Proceedings of the International Symposium on Future Software Technology'97 (ISFST'97). Xiamen: Xiamen University Press, 1997. 399~405
- 13 Lu Ru-qian, Jin Zhi. Knowledge based hierarchical software reuse. Chinese Journal of Advanced Software Research, 1999, 6(1): 1~11

Ontology-Oriented Requirements Analysis

LU Ru-qian JIN Zhi CHEN Gang

(Laboratory of Knowledge Processing and Human Computer Interaction The Chinese Academy of Sciences Beijing 100080)

(Institute of Mathematics Academy of Mathematics and System Sciences The Chinese Academy of Sciences Beijing 100080)

(Laboratory of Management Decision and Information Systems The Chinese Academy of Sciences Beijing 100080)

Abstract In this paper, the authors propose a new approach to requirements analysis, called OORA (ontology-oriented requirements analysis). The most important feature of this approach is that various ontologies have been introduced to represent relationships between objects and/or classes. These ontologies may significantly enhance the expressiveness of the current OOA (object-oriented analysis) approaches. In this paper, first the OORA requirements model is designed. After that, two key types of ontologies are illustrated by two typical use cases. Finally, the main phases in OORA are given.

Key words Requirements analysis, requirements reuse, ontology, information system.