

一种新的语义对象行为语言^{*})

张桂刚¹ Phillip C-Y Sheu^{1,2} 黄蓓蓓¹ 龚志远¹

(武汉大学软件工程国家重点实验室 武汉 430072)¹

(加州大学尔湾分校电子工程与计算机科学系 加州 CA92697)²

摘要 针对人类用自然语言与机器交互所面临的巨大鸿沟,提出了一种基于类自然语言的语义对象行为语言。SOBL是一种宣称式的陈述语言,用于开发数据库驱动的应用,支持复杂的用户交互,主要由一个 SemanticObjectsTM的对象关系开发框架所支持。该语言扩展结构化自然语言,携带语义信息,能够面向非程序员使用,降低了软件的开发难度。最后通过一个具体的实实验证了该语言的有效性。

关键词 语义对象行为语言,语义对象技术TM,语义,语义计算

A Kind of New Semantic Objects Behavior Language

ZHANG Gui-gang¹ Phillip C-Y Sheu^{1,2} HUANG Bei-bei¹ GONG Zhi-yuan¹

(State Key Laboratory of Software Engineering, Wuhan University, Wuhan 430072, China)¹

(Department of EECS, University of California, Irvine, CA 92697, USA)²

Abstract A kind of semantic objects behavior language which is based on natural-like language was given in allusion to the great gulf fixed of human-computer nature interaction by nature language. SOBL is a declarative language for developing database-driven applications as well as supporting complex user interactions. It is supported by an object relational development framework SemanticObjectsTM. It is an extension of the SNL (Structured Natural Language), contains semantic information, and non-programmer can use it, the difficulty of developing software is lowered. At last, the validity of SOBL language was tested and verified via an example.

Keywords Semantic objects behavior language, SemanticObjectsTM, Semantic, Semantic computing

1 引言

让普通用户可以通过自然语言(无论是语音还是打字的方式)与计算机进行交互,是自然语言处理在人工智能领域的主要目标。但是到目前为止,仍然没有这样的人工智能系统能够与人进行自然语言的对话,因为不论是从智能的角度还是语言的角度,研究人员都还没有彻底了解它们的工作方式并制造出这样的人工智能系统。因此目前 NLP(自然语言处理)的主要任务主要集中于处理自然语言的子集,并将该子集应用于有限的知识领域。

目前, NLP 中最大的问题是知识工程的瓶颈问题。要建立一个自然语言理解的机器与人进行对话,就必须建立一个大的公共知识库,但是在日常的交谈中,我们并不是简单地遵从有效的语法规则,而是使用大量的知识来翻译自然语言句子的含义。而且,目前还不清楚这个知识库的核心构件应该是什么?这个库应该有多大?以及如何有效地获得这些大量的领域知识?这是一个典型的人工智能问题,通过读取文档来建立大型的知识库,从而减少知识工程过程中的工作量(减少一个数量级)是 Feigenbaum 提出的计算人工智能的巨大挑战之一^[2]。除了建立一个大型的知识库, NLP 中的另一个瓶颈是“词与语义之间的映射”^[3]。仅仅有知识库和文法规则还不足以理解自然语言句子, NLP 并不仅仅是将单独的英语单

词按照语法规则连接起来就可以构成意义完整的句子。通常的做法是使用某种逻辑作为中间语言来建立语法与语义之间的关联,但是这种方法需要很多规则来告诉机器什么语言模式可以被转换为什么逻辑形式,以及什么逻辑形式可以被转换为知识库中的什么内容。在最坏的情况下,语法与语义之间的每个映射都需要一条规则,而且所有的规则都必须手动去生成。因此,基于这种方法建立的 NLP 系统很难扩展,因为太多的规则要考虑。

上述问题是自然语言句子分析必须经历的各个阶段。语音学、语形学和语法学的研究相对成熟,语义是 NLP 中的一个难点。语义分析过程是 NLP 系统理解自然语言句子意义的过程,而语义构造随着应用领域的不同而有很大的差异,例如,对于飞机订票服务系统来说,语义构造包括航班、航班号、时刻表、价格、订票请求等。对关系型数据库查询来说,每个自然语言语句可以被认为表达了某个 SQL 请求,包括选择、映射和其他关系型数据库操作。因此知识工程的瓶颈问题对于语义分析有很重要的影响。目前处理这种扩展性问题的主要方法包括统计语义角色分配^[4,5]、分类语义角色分配^[6]和增长语义文法^[7]。

本文为了解决自然语言和机器交互面临的困难,以 SemanticObjectsTM^[1] 专利技术为核心,研究一种以它为基础的对象行为语言 SOBL。主要从以下几个方面进行了探讨:首

^{*} 国家高新技术研究发展计划 863 项目(2006AA01Z168)。张桂刚 博士研究生,讲师,主要研究领域为语义计算、语义软件工程; Phillip C-Y Sheu 博士生导师,教授,主要研究领域为语义计算、语义软件工程; 黄蓓蓓 博士研究生,主要研究领域为语义计算、语义软件工程; 龚志远 硕士研究生,主要研究领域为语义计算、语义软件工程。

先详细介绍了 SOBL 语言以及它的文法结构;随后阐述了 SOBL 语言的主要核心技术,主要有语义计算和 SemanticObjectsTM[1] 两个核心概念;接着介绍了 SOBL 语言软件开发方法;第 5 部分用 SOBL 语言编写了一个示例程序并得出实验结果;最后做了总结并提出了今后工作中的展望。

2 语义对象行为语言介绍

2.1 语义对象行为语言

语义对象行为语言(Semantic Objects behavior language, SOBL)是结构化自然语言的扩展。由于自然语言太复杂,很难理解,只有满足一定规格要求的自然语言才能被理解。我们把这种加上了约束限制的、满足语义规则的语言称为 SOBL。

SOBL 是一门用来开发基于数据库的 Web 应用程序的类自然语言编程语言,它支持用户交互以及自定义的复杂工作流,并由对象关系型开发框架 SemanticObjectsTM[1] 所支持。

2.2 语义对象行为语言文法

下面给出 SOBL 语言的文法,该文法是不封闭的,它可以被继续扩充或者修改。

```
SOBL program :- (SOBL-method) +
SOBL-method :- ["entry"] variable-name "(" [type-name variable-name ("," type-name variable-name) * ] ")" "{" method-body "}"
method-body :- (control-structure) * control-structure :- system-action-statement | "If" conditions "Then" "{" (control-structure) * "}" ["Else" "{" (control-structure) * "}" ] | "Loop Repeat" Int "With" "{" (control-structure) * "}" "Until" conditions | "For each" type-name object-name "of" set-value "{" (control-structure) * "}" conditions :- condition (( "and" | "or" ) conditions) *
condition :- object-condition | set-condition object-condition :- object-name method-SNL-expression
set-condition :- "all" "element" "of" set-name method-SNL-expression | "some" "element" "of" set-name method-SNL-expression | "no" "element" "of" set-name method-SNL-expression method-SNL-expression :- ("is" | "are") ["not" | "in" set-name | ("is" | "are") ["not" ] relational-operator object-value
relational-operator :- "greater than" | "greater than or equal to" | "less than" | "less than or equal to" | "equal to"
system-action-statement :- ["If" conditions "then" ] system-action-definition
system-action-definition :- "create" object-name "of" type-name "whose value is" object-value | "create" set-name "of" type-name "whose values are" (set-initialization | set-value) | "let" object-name "be" object-value | "let" set-name "be" set-value | "insert" object-name "into" set-name | "remove" object-name "from" set-name | SNL query | variable-name method-SNL-expression | variable-name "(" [object-name ("," object-name) * ] ")" | java-statements
type-name :- "Int" | "Double" | "String" | "Boolean" | "Char" | "Datetime" | "Vertex" | "Edge" | "Graph" | "set of" type-name
variable-name :- object-name | set-name
object-value :- object-name | object-constant | object-variable
object-constant :- arithmetic-expression | "a new" type-name
object-variable :- object-attribute-name "of" object-value | object-attribute-name "of" set-value
set-value :- set-name | set-constant | set-variable
set-constant :- "empty set of" type-name | set-operation "of" set-value "and" set-value
set-initialization :- "[" object-value ("," object-value) * "]"
```

```
set- variable :- set-attribute-name "of" object-value | set-attribute-name "of" set-value
set-operation :- "intersection" | "union" | "difference"
object-attribute-name :- object-name
set-attribute-name :- set-name
SNL query :- "Find all" object-name "of" set-name "that" method-SNL-expression "call the result" set-name | "Find any" object-name "of" set-name "that" method-SNL-expression "call the result" set-name
arithmetic-expression :- product
plus-operation plus-operation :- ("+" | "-") product [plus-operation]
product :- G multiply-operation
multiply-operation :- ("*" | "/" ) G [multiply-operation]
G :- "(" E ")" |
number-value number-value :- int-value | double-value | object-name | object-variable
```

3 语义对象行为语言核心技术

对象行为语言 SOBL 所依赖的核心技术是语义计算技术^[8]和 SemanticObjectsTM。

(1)语义计算。语义计算所涉及的一些内容在个别学科中可能已经作为独立的单元出现过,如可计算的语言学、人工智能、多媒体、数据库和服务计算。语义计算将这些单元结合成一个完整的系统,并考虑它们之间的协同交互。而且,语义计算不仅仅局限于将信号(如像素、词语)分析和转换成有用的信息,还涉及如何将这些信息通过自然语言或者其它多媒体接口合成和存取,这可以应用到结构化数据和非结构化数据中。语义计算技术可能用来抽取和处理多媒体、文本、服务和结构化数据的内容(语义),如图 1 所示。

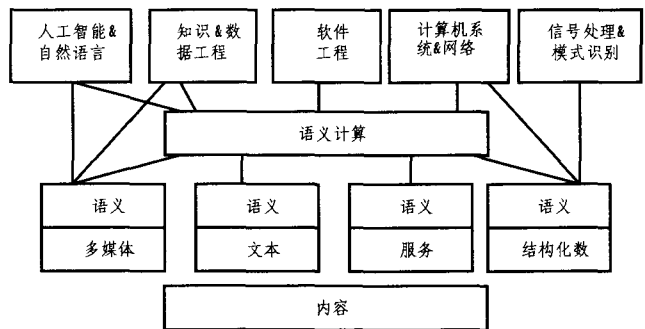


图 1 语义计算技术

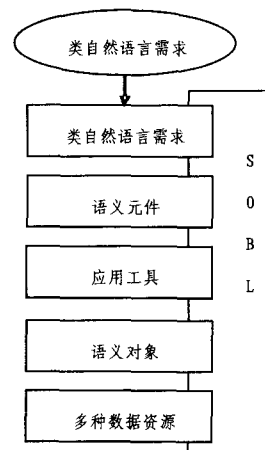


图 2 SemanticObjectsTM结构

(2) SemanticObjects™。SemanticObjects™可以将文件内容转换成对象,也可以将程序或服务的处理能力分解到基础层次,以使用户具有最大的灵活性来组合他们的需求。用户可以用类自然语言来描述他们的需求,如带有一定语法规则的自然语言句子,由名词、动词和形容词组成,这和 LEGO™的处理方式是类似的。这种类似于 LEGO™方法的最大好处就是用户可以将 n 个语义元件组成 $2n$ 个组合结构。SemanticObjects™的结构如图 2 所示。基于对象的统一概念, SemanticObjects™能够实现不同层次数据和相关工具(算法)的无缝整合,而且允许用户像存取层次结构的构件一样存取逻辑对象,不需要管这些逻辑对象实际上是如何存储的。

4 语义对象行为语言软件开发方法

语义对象行为语言是语义软件工程开发方法学^[8]中使用的语义软件开发语言,其主要开发方法流程简介如下:(1)首先用户提出自然语言描述的需求;(2)然后将自然语言需求转变成为 SemanticObjects™能够识别的 SOBL 语言程序即:SOBL 描述的需求;(3)当 SOBL 程序写好后,就通过 SemanticObjects™^[1]进行运行。SemanticObjects™通过 Parser 提取 SOBL 程序里的动词、名词、形容词;(4)当形容词、名词、动词提取出来后,就形成一种能够执行一种以 SemanticObjects™为基础的语义系统的程序即逻辑控制器,这就是一个 SOBL 的语义系统;(5)在执行过程中,由于要满足很多的约束,甚至一些是规则,因此要用到规则引擎来进行规则处理;(6)SOBL 程序验证模块确保程序满足某些特定的属性和条件,并且避免任何设计错误;(7)SOBL 测试模块支持基于详细设计的黑盒测试,它会产生基于参数、参数的定义域和不同测试方法的一个测试案例集合。SOBL 语言应用总体框架如图 3 所示。

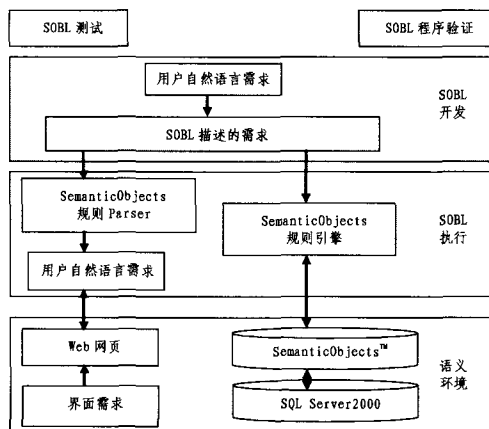


图 3 SOBL 语言软件开发方法

5 语义对象行为语言程序设计应用示例

为了更好地理解如何使用 SOBL 语言进行编程,下面通过一个实例来说明。下面是最小生成树问题的 SOBL 描述。最小生成树,即 Minimum Spanning Tree,它的性质是:设 $G=(V, E)$ 是一个连通网络, U 是顶点集 V 的一个真子集。若 (u, v) 是 G 中所有的一个端点在 $U(u \in U)$ 里、另一个端点不在 U (即 $v \in V-U$) 里的边中,具有最小权值的一条边,则一定存在 G 的一棵最小生成树包括此边 (u, v) 。这个程序目的就是

从一个图形中找出最小生成树。该图形和其权值如图 4 所示。

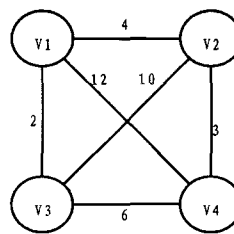


图 4 MST 问题描述中的目标

(1)SOBL 描述如下:

```

entry MST() {
  Create v1 of Vertex whose value is a new Vertex with parameters:
  "V1"
  Create v2 of Vertex whose value is a new Vertex with parameters:
  "V2"
  Create v3 of Vertex whose value is a new Vertex with parameters:
  "V3"
  Create v4 of Vertex whose value is a new Vertex with parameters:
  "V4"
  Create vertexSet of set of Vertex whose values are [v1,v2,v3,v4]
  Create e1 of Edge whose value is a new Edge with parameters: v1,v2
  and 4
  Create e2 of Edge whose value is a new Edge with parameters: v1,v3
  and 2
  Create e3 of Edge whose value is a new Edge with parameters: v1,v4
  and 12
  Create e4 of Edge whose value is a new Edge with parameters: v2,v3
  and 10
  Create e5 of Edge whose value is a new Edge with parameters: v2,v4
  and 3
  Create e6 of Edge whose value is a new Edge with parameters: v3,v4
  and 6
  Create edgeSet of set of Edge whose values are [e1,e2,e3,e4,e5,e6]
  Create g of Graph whose value is a new Graph with parameters: ver-
  texSet and edgeSet
  Create BranchSet of set of Edge whose values are EdgeSet of g
  Create Tree1Set of set of Tree
  Create NodesSet of set of Vertex whose values are VertexSet of g
  Create subTreeEdgeSet of set of set of Edge whose values are subset
  of BranchSet with size 3
  For each set of Edge m of subTreeEdgeSet{
    Create tempTree of Tree whose value is a new Tree with parameters :
    m and VertexSet of g
    If tempTree is a tree then {
      Insert tempTree into Tree1Set
    }
  }
  Select the Tree from Tree1Set whose totalWeight is the smallest
  Call the result mst
  Print the result mst
}

```

(2)执行程序并得出实验结论

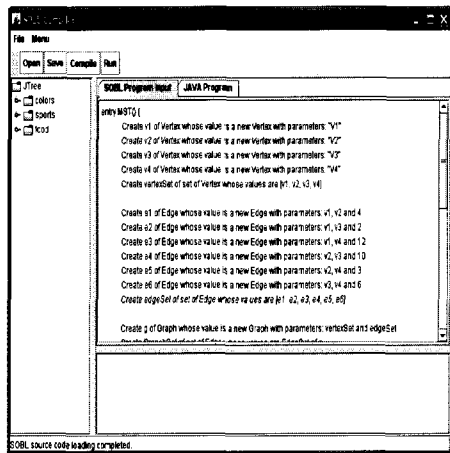


图5 SOBL 程序执行界面

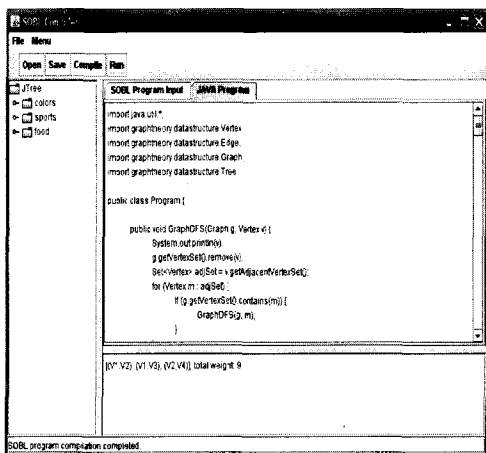


图6 SOBL 程序编译后得出结果

结束语 本文通过以 SemanticObjectsTM为核心技术,并在此基础上研究出一种类自然语言即 SOBL(语义对象行为

语言),来为自然语言和机器进行交互所面临的巨大鸿沟搭起一座桥梁。本文详细介绍了 SOBL 语言的语法结构,阐述了 SOBL 语言的核心技术及它的软件开发方法,并且通过实例验证了 SOBL 语言运行在 SemanticObjectsTM框架中的有效性。结果表明 SOBL 语言是一种较好的面向非专业程序员的类自然语言开发语言,能够大大地简化软件开发的难度。

进一步的工作,主要研究 SOBL 语言作为一种可信编程语言在可信语义软件工程中的具体应用。研究 SOBL 语言在语义服务中作为一种语义服务语言的具体应用等等。

参考文献

- [1] Semantic Objects Getting Started, Copyright 2003. NEC Soft, Co. Ltd.
- [2] Bolognesi T, Brinksma E. Introduction to the ISO Specification Language LOTOS. Computer Networks and ISDN Systems, 1987, 14 (1):25-59
- [3] Drayton L, Chetwynd A G, Blair G S. An Introduction to LOTOS through a Worked Example. Computer Communications (Special Issue on FDTs in Communications and Distributed Systems) Butterworth-Heinemann, 1992, 15(2):70-85
- [4] Ryan M D, Huth M. Logic in Computer Science; Modelling and Reasoning about Systems. ISBN: 0521656028, Cambridge University Press, 2000; 387
- [5] Garcez H E A, Rosenstiel W. CVF-Coverification Framework // Proc. Brazilian Symposium on Integrated Circuit Design. 1998; 103-106
- [6] Strehl K, Thiele L. Symbolic Model Checking of Process Networks Using Interval Diagrams Techniques // Proc. IC-CAD. 1998; 686-692
- [7] Balarin F, Hsieh H, Jurecska A, et al. Formal Verification of Embedded Systems based on CFSM Networks // Proc. DAC. 1996; 568-571
- [8] <http://www.sklse.org/en/en/research1.htm>

(上接第 238 页)

$$T_{\text{设计}} = E_{\text{设计}} \times (1 - Q_3) / n_3 \approx 51.5 \text{ 天}$$

$$T_{\text{编码和单元测试}} = E_{\text{编码和单元测试}} \times (1 - Q_4) / n_4 \approx 28.5 \text{ 天}$$

$$T_{\text{组装与测试}} = E_{\text{组装与测试}} \times (1 - Q_5) / n_5 \approx 78 \text{ 天}$$

最后,根据估算的项目总工期及客户的要求,制定项目进度计划如表 6 所示。

表 6 各阶段、任务的进度计划表

任务名称	持续时间(天)	开始时间	结束时间
任务分析	32.5		
需求分析	80.5		
设计	51.5		
编码和单元测试	28.5		
组装与测试	78		

注:表中所示的时间为有效工作时间。

结束语 本文给出了一种嵌入式软件项目估计方法,为有效地规划和管理嵌入式软件项目、制定合理可行的嵌入式软件开发计划、提高嵌入式软件项目的成功率提供了有力的支持。并结合具体实例对规模估计、工作量/成本估计、关键计算机资源估计和进度估计的内容进行了细致的阐述。

将利用本文方法进行估计的结果,与以往相类似软件实

际结果比较,各项数值的误差均在-15%至+15%之间,并且随着信息量的增加以及不断的细化,估计的误差会逐渐变小,确认了上述方法的实际效用。亚里斯多德曾说:“应该满足于事物的本性所能容许的精确度,当只能近似于真理时,不要去寻求绝对的准确”。估计的基本目标就是产生一个能够在大部分时间内发挥作用的合理的估计。另一方面,计划基于估计,没有估计而制定的开发计划是盲目的计划。因此,我们应尽早估计,经常估计。

参考文献

- [1] Paulk M C, Weber C V, et al. Key Practices of the Capability Maturity Model. SM, Version 1.1. Technical Report, CMU/SEI-93-TR-025, ESC-TR-93-178. 1993
- [2] Pressman R S. Software Engineering A Practitioner's Approach. Fifth Edition. McGraw-Hill, 2001
- [3] Jalote P. CMM in Practice: Processes for Executing Software Projects at Infosys. Pearson Education, Inc., 1999
- [4] 田媛,等. 长春光学精密机械与物理研究所 GJB 5000 过程文件. 中国科学院长春光学精密机械与物理研究所, 2006
- [5] 周汉兵,关昕,马力. 功能点度量在软件开发中的应用. 计算机工程与设计, 2006, 27(3):525-527, 529