

文章编号:1001-9081(2007)09-2327-03

DOM 驱动型智能体在计算任务中的研究与实现

罗云深, 陈志泊

(北京林业大学 信息学院, 北京 100083)

(zbet@sina.com)

摘要:提出了一种能够实现复杂计算任务的以 DOM 驱动模式为基础的智能体实现方法。在复杂计算过程中, 计算模型的调整和修改是一个烦琐的任务, 利用智能体能自动完成此任务, 弥补其中的不足。基本思路是根据智能体的抽象概念逐步实现智能体各基本结构, 并采用 XML 文档存储智能体的组成部分和通信部分, 再利用 DOM 这种驱动模式来读取 XML 的内容, 利用 Script 组件解释该内容, 设计出一个将 XML 技术与智能体相结合的计算架构。该智能体最后应用在森林生长仿真计算任务中, 并取得了较佳预期效果。

关键词:文档对象模型; 智能体; XML; 森林植被模型; 脚本引擎

中图分类号: TP311.52 **文献标志码:** A

Study and implement on DOM-driven Agent in the intelligent calculation tasks

LUO Yun-shen, CHEN Zhi-bo

(College of Information Science and Technology, Beijing Forestry University, Beijing 100083, China)

Abstract: DOM-driven model-based Agent method for implementation of complex calculation was proposed. In some complicated calculation process, there is a need to join some calculation tasks, or a need to adjust and update some calculation tasks for the external environment. In many cases the intelligent calculation can make up for the lack of general calculation. Firstly, basic structure of the Agent was realized progressive in accordance with the abstraction of common Agent. Then storage components and communication segments of the Agent was built by XML document. Here DOM-driven model was used to read the contents of XML. Microsoft Script component was used to translate Agent itself and its actions into local machine code. Then a new framework of a combination of XML and Agent of calculation was created. Finally, as an example, the application of the Agent for calculating forest vegetation simulation gains a desired result.

Key words: Document Object Model (DOM); Agent; XML; Forest Vegetation Simulator (FVS); script engine

0 引言

XML 在实现信息标准化、信息的交流与共享上有其独特的技术优势, 因此它受到了广泛的重视, 几乎所有类型的数据都可以用 XML 来描述。随着互联网和计算技术的发展, 人们的研究兴趣偏向于利用 XML 来进行复杂计算^[1]。在仿真计算任务方面, FVS (Forest Vegetation Simulator) 是目前比较成熟的对森林植被仿真进行计算的软件, 它是美国农业部林务局全国性的预测森林生长和收获模型支撑框架。但是, 在森林生长预测过程中, 生长因子是随环境变化的, 而且部分属于经验公式, 需要随时调整。实际中也存在大量需要动态调整参数和改变运算方法的运算模型, 因此使用这类仿真计算软件存在一些不可避免的问题: 不可重用性、算法改动后需重新编译、交互性能差。

采用智能体来完成这些计算任务可以很好地解决以上问题。DOM (Document Object Model) 驱动型智能体把计算知识存储到知识库中, 用 XML 文档来存储这些知识, 可以方便地根据计算任务提取这些知识, 也可以方便地修改 XML 文档中的知识, 这些知识采用 Script 组件动态解析, 因而不需要再编译了。为说明方便, 文中把这个由 DOM 驱动的智能体一律简称为智能体。

1 智能体所驻留的外部环境

在抽象概念上智能体属于某一环境中能够自主、灵活地执行动作以满意设计目标的行为实体^[2]。智能体不是孤立和封闭的, 它驻留在能够使用 DOM 组件模型的任何软件环境之中, 它需要与以 DOM 为支撑的环境进行持续不断的交互, 它们之间是以 XML 格式文档数据作为信息交换标准。单纯使用 JavaScript 等脚本语言进行动态计算不太灵活, 业务逻辑模糊, 需要一定技巧, 用户难以掌握。而智能体是一种更广泛的计算模式, 可以获取环境中的任何资源, 并具有从环境中学习的能力, 其与环境的关系如图 1 所示。

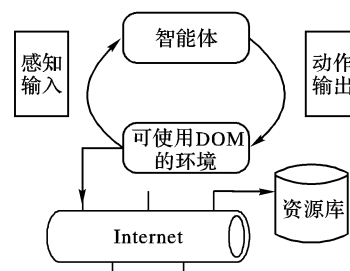


图 1 智能体与外部环境的关系抽象视图

正如 Microsoft office 中的 Agent 需要一个 office 文件环境

收稿日期: 2007-04-27; 修回日期: 2007-07-01。 基金项目: 国家林业局 948 项目 (2005-4-02)。

作者简介: 罗云深 (1978-), 男, 广西平乐人, 硕士研究生, 主要研究方向: 数据库、计算机软件与理论; 陈志泊 (1967-), 男, 山东莒人, 副教授, 博士, 主要研究方向: 数据库、计算机软件与理论。

一样(如 Word),这里的智能体采用 VC 调用 MSXML 命名空间,声明一个 DOM 对象,产生一个客户端可执行的环境。智能体根据从环境中获得的感知输入,自主地决定该执行什么样的动作以实现其设计目标。这里的感知输入在计算任务中表现为用户对计算模型的请求,该请求方式可由如下 DTD (Document Type Definition) 结构定义:

```
<!DOCTYPE Calculation [
  <!ELEMENT Calculation (形参表,实参表,公式名,公式体,运行形式) >
  <!ELEMENT 形参表 (#PCDATA) >
  <!ELEMENT 实参表 (#PCDATA) >
  <!ELEMENT 公式名 (#PCDATA) >
  <!ELEMENT 公式体 (#PCDATA) >
  <!ELEMENT 运行形式 (单独|批量) >
]>
```

而在这里的设计目标是:智能体能够组织不同的计算实体完成用户指定的计算任务。根据这个设计目标,智能体不仅需要能感知用户的计算请求,在不同层次中能够监控运行计算状态^[3],并对用户的输入进行分析以判断它是否有误。当智能体发现用户的输入是错误的,或者根本找不到对应的请求公式,那么它将使用已有的计算知识来完成用户给出的请求,也可以对用户进行一定提示。在智能体中可以使用 try、catch 等捕获异常的解釋型脚本语句来判断非正常输入。

2 智能体的行为实体设计

智能体为了实现其设计目标,它具有一组动作并能执行这些动作,是动作的决策者和实施者,可以把它看成一个行为实体。在这里可以定义两个类来组织智能体的动作,而这些动作主要体现在:为智能体加入新的计算方法,为智能体修改替换已有计算方法,给出计算信息。通过这些动作的实施,也许可以改善计算任务的执行,也许让计算任务变得更糟。这两个类分别为计算节点类和计算容器类,下面给出它们具体的 JavaScript 脚本描述方法:

```
function Calculation_node ( name, key, text, info, paras ) {
  /* 计算节点类 */
  this.名称 = name;
  this.关键字 = key;
  this.内容 = text;
  this.说明信息 = info;
  this.版本信息 = version;
  this.计算参数 = paras;
  this.内部运行 = new Function();
}

function Calculation_collection () {
  /* 计算容器类 */
  this.增加计算方法 = addFunc;
  this.计算方法名称 = itemWithName;
  this.已有计算方法 = itemNameExists;
  this.计算体位置 = posOfItem;
  this.长度 = 0;
  return this;
}
```

3 智能体的自主性

自主性是智能体的基本属性,在计算任务过程中,智能体无需用户指定计算请求和计算模型的对应关系。智能体拥有一个基本的计算模型库,并可通过前面的计算容器类可以不断加入新的计算方法,让智能体得到不断的学习。一旦感知

到用户对计算模型的请求,智能体将首先判断输入部分是否正确,然后通过基本计算模型库中的查找,将对应的可执行运算过程找出来,自动加载到智能体的执行指令中,最后通过 Script 组件对象对该指令进行解释,输出最终结果。该过程如图 2 所示。

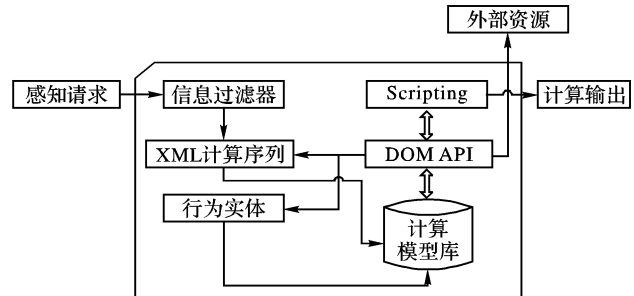


图2 智能体处理器结构

信息过滤器是处理异常的一段脚本。因为智能体是基于 XML 格式的,所以首先要判断请求格式是否正确,在智能体中使用了 DOMDocument 的 xml 属性来判断请求格式,其次通过前面提到的异常处理脚本方法即可对其他非法信息进行过滤。可以通过如下 DTD 定义 XML 计算序列结构:

```
<!DOCTYPE getRemoteFun [
  <!ELEMENT getRemoteFun (url, arguments) >
  <!ELEMENT url (#PCDATA) >
  <!ELEMENT arguments (#PCDATA) >
  <!ATTLIST url error-hint CDATA #IMPLIED >
  <!ATTLIST arguments length CDATA "0" >
  <!ENTITY CALLBY "spFuncdso" >
]>
```

其中,spFuncdso 实体的声明如下:

```
spFuncdso = new ActiveXObject("MSXML.DOMDocument");
```

arguments 是 JavaScript 语言中任何函数体外部参数指定的默认接收变量,其长度由 length 指定,默认为 0,在输入时每个传递常用", "来分割,在脚本中以数组形式存在。计算模型库的结构较为简单,以 CDATA 数据格式存储运算过程,其 DTD 定义简要如下:

```
<!DOCTYPE data [
  <!ELEMENT data (#CDATA) >
  <!ATTLIST data name CDATA "Fvs" >
  <!ATTLIST data tagum CDATA "datagroup/DataRendering" >
  <!ATTLIST data argv CDATA "o" >
  <!ATTLIST data driver CDATA "RequestDriver" >
  <!ENTITY CALLBY "spFuncdso" >
]>
```

要让 DOM 层读出 XML 计算序列,需要额外加载一个 Script 组件对象,在 VC 中利用 Microsoft Script 组件使用户能创建运行任何 ActiveX scripting 引擎。通过该引擎申明一个或多个全局智能体对象,除包含前面提到的行为实体外,还包含该智能体的私有信息和接口方法。这个智能体是一个由 JavaScript 描述类,属于解釋型智能体。其中,spFuncdso 行为实体是它的一个重要属性,通过 spFuncdso 继承 IDOMDocument 接口的 load 方法,可以读取网络中任何可用的 XML 数据资源。智能体的另一重要属性就是 getRemoteFun,在 JavaScript 解释机制中,它是一个可接受动态参数的函数体,在其内部直接使用了 spFuncdso 行为实体。通过智能体中计算容器类的增加计算方法,将该属性加入到行为实体中,那么智能体就可以识别该属性的各项特征,实现

了智能体对本身的自描述。getRemoteFun 函数体的核心处理伪代码如下:

```

spFuncdso.async = false;           //非同步模式
spFuncdso.load( arguments[0] );    //第一个参数(url)
检查 spFuncdso.xml 是否错误类型;
计算容器类中加入参数指定的计算方法;
return;

```

4 智能体之间的合作计算

在计算任务中,可能某些复杂的计算需要分解^[4],使得业务逻辑更清晰,理解更简单。不同的智能体包含不同的计算方法,它们之间可以通过 spFuncdso 获取对方的信息,并可根据需要与对方进一步交互,以判断是否执行对方的计算方法,或者放弃使用,其过程如图 3 所示。

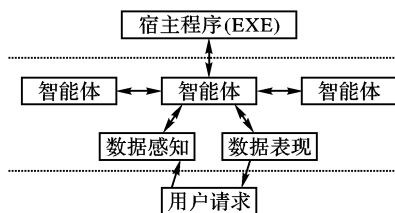


图 3 智能体关系图

可以看出,如果只是单纯使用 XML 来存储需要定义的数据标志,那么它本身是孤立的、静止的、离散的。智能体在利用 XML 作为数据交换标准后,它们之间是关联的、动态的、计算连续的。因为前面提到智能体可以对自身进行描述,才为智能体之间相互识别对方提供可能,甚至可以激活网络中未被启动的智能体。图 3 中的宿主程序为任意可以同时使用 DOM 组件和 Script 组件的程序,这些程序可以是同一种语言开发的,也可以是不同语言开发的。通过智能体网络间的相互合作^[6],使得智能体能够更好地实现其计算任务的设计目标。

5 在森林生长仿真计算中的框架

5.1 计算框架的构造

为计算出适合我国某地区的森林生长仿真结果,将 FVS 中计算材积等 273 个计算知识写入计算知识库中,根据前面的智能体模型,构造出图 4 所示的计算框架。把不同地区数据和地理特征要求分别输入到林分调查数据和预测请求中,由该智能体来完成整个计算任务。

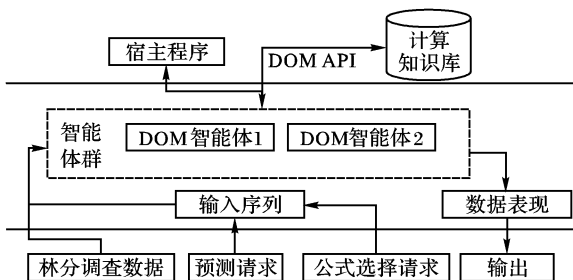


图 4 智能体在森林生长仿真计算中的框架图

在计算任务中,设原始数据输入序列为 $\{D_1, D_2, \dots, D_n\}$, $D_i (i = 1, \dots, n)$ 代表一行输入数据,如表 1 中行所示,输出序列为 $\{O_1, O_2, \dots, O_n\}$, $O_i (i = 1, \dots, n)$ 代表一行输出数据,如表 2 行所示。计算任务的序列为 $\{T_1, T_2, \dots, T_n\}$, 在每个 $T_i (i = 1, \dots, n)$ 中,可包括多个计算 Calculation 结构:

$$T_i = \{Calc_1, Calc_2, \dots, Calc_n\}$$

表 1 数据输入表

Plotid	Treedid	code	dbh	dbhinc
7906	1	LP	5.0	1.5
7906	2	LP	9.5	2.0
7906	3	LP	4.5	0.6
7906	4	LP	3.2	1.3

注: Plotid—立地编号 Treedid—树木编号
code—树种代码 dbh—胸径
dbhinc—胸径增长量

令计算容器为 $C, C = \{\{T'_1, T'_2, \dots, T'_n\}, \dots, T'_i\}$ ($i = 1, \dots, n$) 表示对 T_i 的处理;令程序开始计算部分为 start, 则有:

$$start = C' = \{\{T'_1, T'_2, \dots, T'_n\}, \dots\}$$

其中: $C' \in C$ 。最后输出序列规则如下:

$$\{D_1, D_2, \dots, D_n\} \wedge \{T_1, T_2, \dots, T_n\} \wedge C' \rightarrow \{O_1, O_2, \dots, O_n\}$$

其中: C' 必须包含一组对 $\{T_1, T_2, \dots, T_n\}$ 的操作,并可包含多组其他操作,实现复杂计算的多样化处理。在实现过程中, C 通过 addFunc 方法加入每个 T'_i 处理过程, $T'_i = getRemoteFun(T_i)$ 。在森林生长仿真计算任务的序列中, T_m 为一个典型的计算(预测单木树高),首先为它加入一个 Calculation 结构,公式体如下:

$$treeht[j] = 1.3 + Exp(2.367 - 15.931 \times (1/(dbh[j] + 1)))$$

其中:把 $[j]$ 作为批量运算的书写规则,表示对数组批量处理,同时把运行形式设置为批量。在 C' 加入两个 T'_i 处理序列,一个处理输入变量的接受逻辑,一个处理输出格式定位的业务逻辑。最后执行智能体的宿主程序,激活该智能体,得到表 2 输出结果。

表 2 数据输出表

plotid	treedid	code	dbh	dbhinc	treeht
7906	1	LP	5.0	1.5	4.63
7906	2	LP	9.5	2.0	6.96
7906	3	LP	4.5	0.6	4.26
7906	4	LP	3.2	1.3	3.16

5.2 计算框架的实现

在计算实现时,把调查数据和各种请求按行记录格式写入 XML 文件中,实验中把这个文件命名为 treedata,来记录输入请求。再在 XML 中定义一个 CDATA 节点,其中用脚本定义若干个数组来记录输入和输出序列,命名为 exec 接口文件。计算模型库分为基本计算和扩展计算,指派 function 文件来存储前者,定义一个计算容器文件 cacl 存储库中计算的批处理调用形式,形成所有计算序列操作;计算序列 T 采用函数调用形式,但每个函数名是由前面计算模型库结构中的 name 决定的。那么请求中只判断执行 C' , 即部分计算。所有这些计算模型和计算容器就形成了计算知识库,前面定义计算序列的 url 就是指向计算模型库文件的。

在处理输入时,采用函数 selectSingleNode 和 split 把行记录提取到数组中。实验中宿主程序寻找执行接口文件 exec,在其中通过 getRemoteFun 方法指定读取 treedata 中的数据和操作,完成整个计算过程。 (下转第 2333 页)

表达方法较好地解决了基因序列图形表达中的退化问题,同时这类表达还保留了序列的生物学特征。在这类表达的基础上应用聚类分析对 H5N1 病毒基因序列进行分析,在聚类方法中引入 *PFS* 参数来求证最佳的聚类结果,同时这种分析方法中直接利用基因序列的数字特征值作为分析原始数据,减少了相似矩阵的建立从而减少了聚类分析中的复杂的数据计算。分析结果表明利用文中所给图形表达建立基因序列数字特征矩阵进行的聚类分析具有一定的合理性。

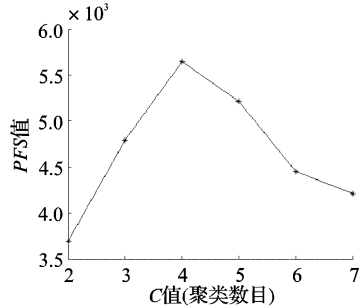
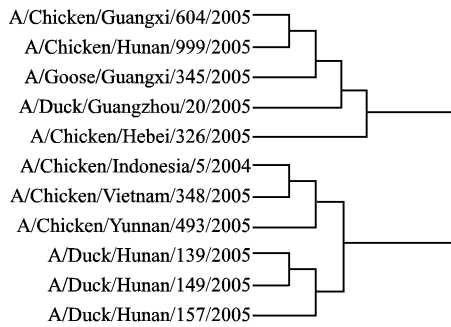


图1 PFS 随聚类数目变化



参考文献:

- [1] ZHANG C T, ZHANG R. Analysis of distribution of bases in the coding sequences by a diagrammatic technique [J]. *Nucleic Acids Research*, 1991, 19: 6313 - 6317.
- [2] ZHANG R, ZHANG C T, CURVES Z. an intuitive tool for visualizing and analyzing DNA sequences [J]. *Journal Biomolec Struct Dyn*, 1994, 11: 767 - 782.
- [3] GUO F B, OU H Y, ZHANG C T. ZCURVE: a new system for recognizing protein coding genes in bacterial and archaeal genomes [J]. *Nucleic Acids Research*, 2003, 31: 1780 - 1789.
- [4] ZHENG W X, CHEN L L, OU H Y, *et al.* Coronavirus phylogeny based on a geometric approach [J]. *Molecular Phylogenetics and Evolution*, 2005, 36(2): 224 - 232.
- [5] YUAN C X, LIAO B, WANG T M. New 3D graphical representation of DNA sequences and their numerical characterization [J]. *Chemical Physics Letters*, 2003, 379: 412 - 417.
- [6] RANDIC M, ZUPAN J, NOVIC M. On 3-D Graphical representation of proteomics maps and their numerical characterization [J]. *Journal of Chemical Information and Computer Sciences*, 2001, 41(5): 1339 - 1344.
- [7] RANDIC M, VRACKO M, NANDY A, *et al.* On 3-D Graphical representation of DNA primary sequences and their numerical characterization [J]. *Journal of Chemical Information and Computer Sciences*, 2000, 40(5): 1235 - 1244.
- [8] LIAO B. A 2D graphical representation of DNA sequence [J]. *Chemical Physics Letters*, 2005, 401: 196 - 199.
- [9] LIAO B, TAN M, DING K. Application of 2-D graphical representation of DNA sequence [J]. *Chemical Physics Letters*, 2005, 414: 296 - 300.
- [10] LIAO B, WANG T M. Analysis of similarity/dis-similarity of DNA sequences based on 3-D graphical representation [J]. *Chemical Physics Letters*, 2004, 388: 195 - 200.
- [11] LIAO B, SHAN X Z, ZHU W, *et al.* Phylogenetic tree construction based on 2D graphical representation [J]. *Chemical Physics Letters*, 2006, 422: 282 - 288.
- [12] LIAO B, DING K. A 3 D graphical representation of DNA sequences and its application [J]. *Theoretical Computer Science*, 2005, 358(1).
- [13] ZHANG Y S, LIAO B, DING K. On 2D graphical representation of DNA sequence of nondegeneracy [J]. *Chemical Physics Letters*, 2005, 411: 28 - 32.
- [14] 骆嘉伟, 李仁发, 张白妮. 基于多维伪 F 统计量的基因表达动态聚类分析方法研究 [J]. *系统仿真学报*, 2006, 18(3).
- [15] HRUSCHKA E R, CAMPELLO R J G B, De CASTRO L N. Evolving clusters in gene-expression data [J]. *Information Sciences*, 2006, 176 (13): 1898 - 1927.
- [16] BALDESSARI D, SHIN Y, KREBS O, *et al.* Global gene expression profiling and cluster analysis in *Xenopus laevis*. *Mechanisms of Development*, 2005, 122(5): 441 - 475.
- [17] TSANG K W, HO P L, OOI G C, *et al.* A cluster of cases of severe acute respiratory syndrome in Hong Kong [J]. *The new england journal of medicine*, 2003, 348: 1977 - 1985.
- [18] 陈希孺. 数理统计引论 [M]. 北京: 科学出版社, 1981.
- [19] PHYLLIP [EB/OL]. [2007 - 02 - 10]. <http://evolution.genetics.washington.edu/phylip.html>.

(上接第 2329 页)

6 结语

智能体的发展已经到了一个阶段,但其研究大多数为理论研究^[8],实际应用较少。本文通过从智能体的抽象概念入手,采用 DOM 这种读写 XML 文件的驱动模式,结合智能体的基本性质和特征,开发了一个专门代理计算任务的解释型的智能体模型。该智能体应用在森林生长仿真计算任务中,只需补充 XML 文档类型的计算知识到知识库中,便可适用于各地区的计算要求,毋须重新编译便可执行,极大减少了计算编码工作和人工操作。而智能体的通信协议和标准化、外部接口的协议和标准化,以及心智状态等,都是下一步要研究的方向。

参考文献:

- [1] 彭涛, 张力. 基于本体和 XML 的数据交换研究 [J]. *计算机工程*, 2006, 32(1): 90 - 92.
- [2] 毛新军. 面向主体的软件开发 [M]. 北京: 清华大学出版社, 2005.
- [3] 孙振平, 贺汉根. 基于层次式的实时智能计算体系结构研究 [J]. *计算机工程与科学*, 2006, 28(4): 136 - 138.
- [4] 黄德双. 智能计算研究进展与发展趋势 [J]. *中国科学院院刊*, 2006, 21(1): 46 - 52.
- [5] 吉根林, 韦素云, 鲍培明. 一种基于 DOM 树的 XML 数据频繁模式挖掘算法 [J]. *南京航空航天大学学报*, 2006, 38(2): 206 - 211.
- [6] 吴文昭. 信息的智能化技术综述 [J]. *甘肃政法成人教育学院学报*, 2006(1): 179 - 180.
- [7] BUTZ C J, HUA S, MAGUIRE R B. A Web-based bayesian intelligent tutoring system for computer programming [J]. *Web Intelligence and Agent Systems*, 2006, 4(1): 61 - 81.
- [8] LUCK M, MCBURNEY P, GONZALEZ-PALACIOS J. Agent-based computing and programming of Agent systems [EB/OL]. [2007 - 02 - 01]. <http://www.dcs.kcl.ac.uk/staff/mml/publications/assets/promas06.pdf>.