

基于 KQML 语言的医学图像 agent 协调系统设计^{*}

The Medical Image Agent Collaboration System Implementing Based on KQML

张全海 施鹏飞

(上海交通大学图像处理与模式识别研究所 上海200030)

Abstract The applications on web have three important characters: distributed of data and control, information uncertainty and environment dynamism with the development of the network technology. Taking care of the huge computation of the medical image processing. To realize the collaboration, the property of the KQML is analyzed firstly, then the architecture of the collaborated agent in the MAS environment is designed, and the state, performatives, conversation of the collaborated agent are described. The collaboration architecture can combine with the CORBA organically and have the properties such as open-ability, intelligent. It can suit to the application on the Web in the future.

Keywords Agent, Collaboration, KQML

1. 引言

生物技术、计算机技术和数字图像技术的发展使医学图像技术数字化和网络化,多模式诊断图像如 CT、MRI、PET、SPET 和超声图像的出现和应用,使现在医学诊断和护理治疗得到了质的发展。医学图像处理本质上是一种应用算法(也称作图像滤波器)来强化和分析医学图像的计算技术。这些诊断图像尺寸都很大,处理起来需要消耗较大的计算资源,不适合单机处理。随着网络技术的发展,网络计算已经成为一种正在兴起的计算模式。而传统计算中的封闭性,即只能利用本地资源,对于医学图像处理而言,需要综合更多的知识,因此需要分布的和基于网络的处理。为了充分利用网络计算资源,需要进行分布式计算处理并加强计算体之间的协调交互。近年来,分布式人工智能(DAI)技术的发展为解决分布对象之间的协调操作提供良好的理论基础。分布式人工智能是研究人工智能计算中的开放性和交互性,其核心就是强调系统中计算的协调性,它将传统的通用问题求解策略和基于知识表达的基本推理形式向面向对象的混合协作的模式发展,由集中解决向分布解决发展^[1~4]。分布式人工智能和知识系统有利于不确定状态下的医学临床诊断以及医学图像处理和分析中信息的获得,为了实现各计算体之间的协调交互性,将医学图像处理向当前的网络计算模式发展,在上层应用中采用分布式人工智能中的 Multi-Agents (MAS) 机制,底层采用 CORBA 进行分布应用。通过 MAS 机制中一组逻辑上或物理上分布的智能软件 Agent 对象间的交互和协商,使全局问题的求解合理、有序。MAS 具有智能性、自适应性和层次性,能有效实现系统的重组和协调。在 MAS 中,Agent 对象间的协调工作需要具备三个因素:agent 通讯、agent 设计和系统结构建模。知识的获取和传递对于 agent 对象间的协调是关键的因素,这些工作的完成就需要 KQML (Knowledge Query and Manipulation Languages) 来实现。

KQML 语言是一种软件 agent 间的通信语言,利用 KQML 可以实现 agent 间的信息交换和知识的实时共享^[5,6]。KQML 语言定义了一系列可扩展的行为原语集(per-

formatives) 规范了 agent 间获取信息和知识的行为操作。本文以 KQML 语言为基础,对 agent 协调进行了理论研究,通过对 KQML 语言行为原语的扩展构造了适合网络计算的医学图像协调处理系统,定义了适合协调处理的行为原语和会话策略,该协调系统具有开放性、智能性和资源可重用性,能满足协调处理的要求。

2. KQML 语言

知识查询与处理语言 KQML 是由美国国家研究计划署 (ARPA) 提出的知识共享计划 (Knowledge Sharing Effort, KSE) 的一部分,该计划主要是构造大规模的可共享和可重用知识库,KQML 语言充当不同知识表示语言的中间载体,使 MAS 环境中 agent 实现知识共享。KQML 作为一种信息格式和信息处理的协议,主要用于各 agent 间的知识实时共享和行为协调,为 agent 间和其它的信息服务器和客户之间提供了一种标准的语言规范。KQML 不仅提供了通过一种特殊的用于 agent 的交互协调的功能集合,称之为调节器 (facilitators) 的知识共享的基本结构,而且定义了一系列可扩展的行为原语集,用于构造更高级别的 agent 交互模型。

2.1 KQML 语言模型

KQML 语言模型由三个部分组成:通信层、信息层和实体层,如图1所示。通信层主要进行信息的编码,对基本的通信参数进行描述,如发送者和接受者的地址标记等,该层建立在进程间通信 (IPC) 或者 TCP/IP 等基本传输协议之上,并且独立于这些传输机制 (TCP/IP, SMTP, IIOP 等)。信息层主要用于定义应用中相互间传送的信息,它形成了 KQML 语言的核心,决定了 agent 间交互的类别。信息层的主要功能还包括识别被用来发送信息的网络协议,提供行为原语与原语实体的连接。原语实体描述语言的识别和解释以及分析、推理等功能的选择执行,实体层包含每一个消息和原语的具体内容以及表示内容所采用的知识语言如 LISP 等。KQML 的内容描述可以由任何一种知识表示语言描述,可以是使用 ASCII 字符的表示语言,也可以是使用二进制的表示语言,这体现了 KQML 语言的开放性。当使用 KQML 时,软件 agent 将传输

^{*} 本项目得到自然科学基金项目 (No. 69931010) 资助。张全海 博士研究生,主要研究方向:图像分析,网上医学图像三维重建。施鹏飞 博士生导师,教授,主要研究方向:图像处理,模式识别,人工智能。

的内容封装在 KQML 消息中,接受机构不需要辨认这些部分,而只辨认消息的开始和结束。

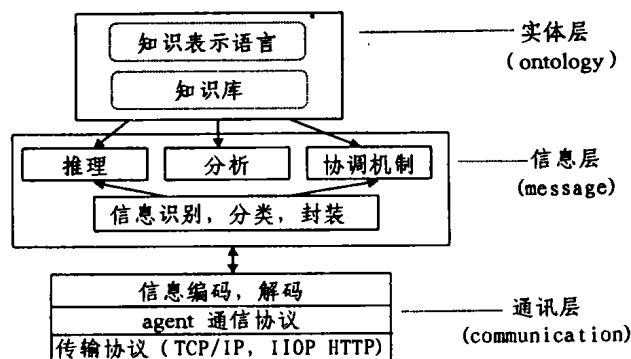


图1 KQML 语言模型

2.2 KQML 语言语法和原语

KQML 的语法相对其它知识表示语言来说,由一对括号体序列组成,括号体中第一个元素是原语和相关原语参数,行为原语的参数是可选的,在调用参数时必须在前面注明关键字,关键字以冒号(:)标记,原语的顺序可以交换。KQML 的语法类似 LISP 语言,但比 LISP 具有较强的灵活性和扩展性,消息表示简洁,如下所示:

```
(query / * 原语标识
:sender joe / * 发送者
:content (PRICE IBM ?price)
/* 原语执行内容,price 为变量
:receiver stock-server / * 接受者
:reply-with ibm-stock / * 应答的标识变量
:language LPROLOG / * 实体描述语言
:ontology NYSE-TICKS / * 本语定义集合)
```

与 KQML 语言模式相对应,关键字 content 内容用于实体层,reply-with, sender, receiver 内容用于通信层, language, ontology 内容用于信息层。

在 KQML 语言中定义了一些基本的用于信息通信的原语,这些原语可以被 agent 利用来构成更复杂和抽象的协调策略。原语实际上就是发送消息的行为,具体内容封装在消息中, KQML 基本原语可以分为8类。

2.3 KQML 软件体系结构

KQML 软件体系结构为 agent 协调和管理以及进行程序设计提供了一个参考模型,如图2所示。在软件体系结构中,

关键的是2个部件:路径选择器(router)和 KRIL (KQML Router Interface Library)。router 为每一个 agent 提供了一个应用程序与网络的界面接口,实现网络连接,如为 agent 应用提供客户/服务器的函数调用,管理与其它 agent 建立连接等,能自动处理基于 KQML 的信息交互,这些交互依赖于 KQML 原语和其相关参数的设定。KRIL 是 agent 应用的人机界面,为 agent 应用程序与 router 提供编程接口,设定参数和路径,建立信息连接并进行 KQML 消息的分析,为 KQML 的消息内容语言无关性提供保证。

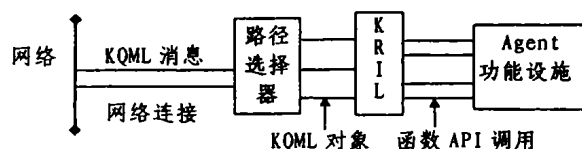


图2 KQML 软件体系结构

3. 协调 agent 结构设计

KQML 语言的提出是为了更好地实现 agent 间的事务协调处理,agent 是协调操作的主体,因此在 MAS 系统中,应对 agent 进行合理的设计使之能有效地完成协调操作。对 agent 进行设计应当处理好以下问题:(1)agent 的内部状态和组织形式的明确;(2)agent 逻辑推理和协调策略的制定。对 agent 设计最常用的模型是 BDI (Belief, Desire, Intention) 模型,本文以 BDI 模型为基础对协调 agent 结构进行了设计。按照 BDI 模型,agent 结构应该由多层知识库,工作空间,通信服务器,任务选择、分解、分配模块和人机交互界面构成,其逻辑结构图如图3所示,其中反应模块可以选择使用,该模块用于对新数据的到来,旧数据的更新和其它 agent 消息的提交向系统作出适时的反应。Agent 的协调机制,会话策略以及推理机制包含在知识库中,因而多层知识库可以细分为基于 agent 所在域的知识,控制知识和逻辑推理知识构成。逻辑规则包括协调、决策、学习,任务管理规则工作空间主要存储控制信息,临时交换信息和其它共享信息如各 agent 状态信息、任务分配信息等,作为 agent 交互的共用空间类似于全局的白板通信服务器负责信息的合成和分解、分类、判断。图形界面则建立 agent 与用户之间的信息传递。

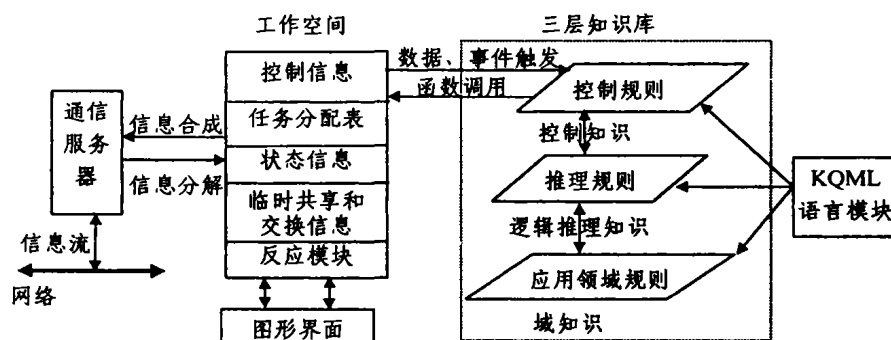


图3 协调 agent 结构

4. 协调处理系统逻辑设计

整个协调过程分为8个状态,状态变迁的有限状态机图如图4所示,状态采用有限状态机定义。

状态1 初始状态,即 agent 处于待命状态,准备就绪可以接受任务。

状态2 目标分析状态,agent 分析申请的任务,如果可以本地完成则接受进入状态4,如果不能本地完成,拒绝任务申

请,如果能部分完成,则向调节器发送请求,希望能协调完成,进入状态3。

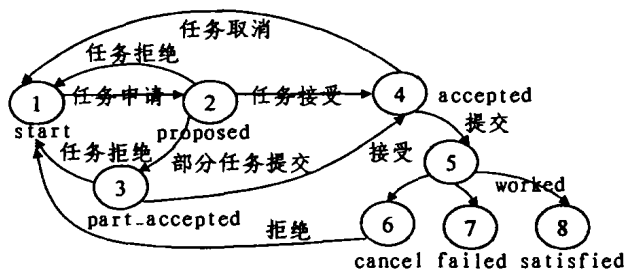


图4 agent 状态变迁

状态3 等待调节器结果,如果没有其它 agent 能完成或者部分完成以及 agent 拒绝协调,则拒绝,否则接受任务,进入状态4。

状态4 任务接受状态,如果协调 agent 有一个没准备就绪则任务不能执行,返回状态1;全部就绪,则任务提交,进入状态5。

状态5 任务处理状态,各协调 agent 通过协作,对任务进行处理。

状态6 任务在处理过程中被取消后 agent 处于的状态。

状态7 任务处理失败状态。

状态8 任务处理成功状态。

根据状态图,为了实现协调处理,在 KQML 原语基础上,引入了以下 agent 行为原语: request, accept, reject, query, cancel, acknowledge, alter, fail, commit, inform 和 propose 等。在使用 KQML 的 MAS 中,在每个 agent 域中至少有一个称作 facilitator 的特殊 agent 负责处理网络中 agent 的注册,用于查找相关的 agent。原语中, propose 用来向 agent 提交处理的任务, accept 和 reject 用来对提交的任务作出反应,本地 agent 无法完成任务时,使用 request 向 facilitator 请求其它协调 agent, cancel 用于处理任务的 agent 取消正在处理的任务, acknowledge 用于对消息发送的验证信息, inform 用于 agent 间传送数据和信息, alter 用于对数据进行更新和改变, query 用于 agent 间信息的查询, commit 和 fail 用于 agent 向任务所有者通告任务的执行结果。各个行为原语严格按照 KQML 语言规范进行定义,原语的执行体可用实体描述语言进行描述。

在 MAS 系统中,各 agent 的协调是通过一系列的会话

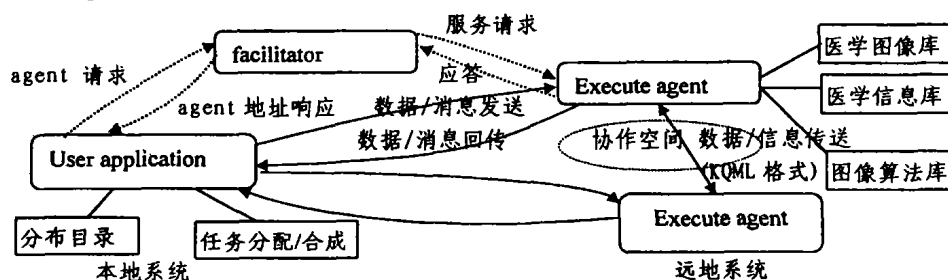


图5 协调系统通信示意图

在实际的医学图像协调设计时,选择 CORBA (common object request broker architecture) 或 DCOM 等分布式软件作为异构网络和机器的连接,因为 CORBA 解决了分布式系统之间的互操作问题。在 CORBA 的基础上,可选择 Java 或 C++ 等支持 CORBA 的工具开发应用程序,利用 Java 的多线

(conversation) 来实现的,在会话系列中, agent 可以按照相互达成的规则进行信息交换,状态变迁和执行本地行为。会话是 agent 间同一交互过程中的 KQML 消息序列,会话策略与特定的应用有关,不同的 agent 有不同的会话策略,但总体上包括以下三个部分:会话初始化、会话进行(连接、持续和断开)和会话同步。会话由一系列的 agent 状态和规则组成, KQML 语言定义如下:

```
(def-conversation 'client-conversation
  :name 'client-conversation
  :content-language 'list
  :speech-act-language 'kqml
  :initial-state 'start
  :final-state '(cancel, failed, satisfied)
  :rules '((start r1) (proposed r2) ((accepted
    r3) (part-accepted r4) (worked r5)))
```

其中 r1 到 r5 为具体的会话策略,描述了当会话在一定状态时比如 start 状态时可能采取的相应措施,根据 agent 的状态图,例如当 client agent 处于 start 状态时,采取 r1 策略,进入 proposed 状态,采取 r2 策略进入 accepted 状态。会话策略的 KQML 语言定义类似会话类的定义。

5. 协调处理系统的实现技术

医学图像的 agent 协调处理其涵义是利用 agent 来实现全局处理任务的动态分配,网络资源,处理信息和功能的动态选择及共享,通过任务实现过程中各 agent 间的信息的交互、讨论,消解矛盾和冲突来实现任务的协调,因此由于各 agent 自身的复杂程度不同, agent 间数据的交换与共享,实现信息交流是协调的主要部分,如何寻找合适的 agent 实现资源动态的组合也是该系统需要解决的核心问题。通信是系统实现需首要解决的部分,图5是协调系统的通信示意图,可以实现 agent 的动态组合。为了减少用户程序对相关代理的查找,系统中加入协调者 (facilitator),其实际是一个智能 agent 的管理器,负责建立用户程序与实现 agent 之间的连接,如果用户知道实现 agent 的网络位置,则可以直接与该 agent 建立通信连接,否则就需要通过 facilitator 来实现, facilitator 上面保存着网络中的 agent 动态分布目录,当新 agent 加入或旧 agent 离开时候都要到 facilitator 来登记。在用户看来,通过 facilitator 实现与 agent 连接是透明的。Facilitator 的加入有效地扩大了网络服务的范围,并能实现 agent 的动态组合。各个 agent 之间可以按照 KQML 语言的规范来相互传递信息,以实现资源、知识和算法执行的相互合作和密切配合。

程机制编写 agent 协调代码,以及对 agent 设计,构造 agent 类库。平台选择 Windows NT,利用 Windows 的图形界面开发人机交互界面,Java 语言具有很强的移植性和可重用性,网络编程的特点,因此利用 Java 语言来编写 agent 协调代码并结合 KQML,可以在 agent 之间传递高层的、抽象的概念知

识和信息交换,Java 和 CORBA,MAS 结合可以保证医学图像协调系统的实现,软件体系结构如图6所示。

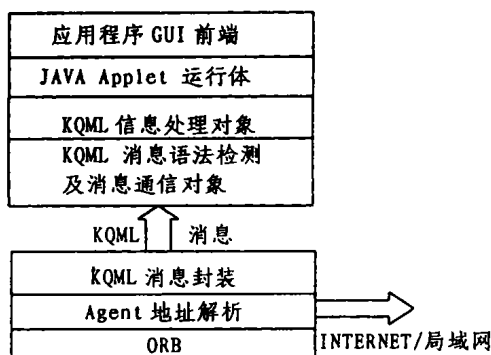


图6 协调 agent 软件体系逻辑结构

6. 医学图像分割协作的实现

医学图像在辅助医生对病症的诊断和临床实践以及医学研究方面上具有很重要的意义和广阔的应用前景,三维医学图像的发展已有20多年的历史,从三维图像数据的重建、显示到分析方面的研究,都做了大量工作。对于三维图像分析方法的研究,一个最基本且重要的问题就是图像分割方面,在单机上进行分割,已经产生了很多有意义的算法,如区域分割和表面边界分割,但是各个算法各有特色,针对的应用范围也各不相同,而且医学图像往往具有复杂的结构,包括不同的组织,如骨骼、软组织等,由于各种图像信息具有互补性质,通常通过对同一个部位的不同模式的医学图像信息进行集成来获得临床应用中所需要的全面信息,需要对不同的组织和图像模式采用不同的分割算法进行处理,这就需要网络协作处理技术。本文利用软件代理的分布式协作结构及 KQML,对三维医学图像的协作分割进行了算法设计,协作的流程如下:

1. 由用户界面输入欲分割图像或通过界面从网络浏览得到,并且输入分割目标及评价指标。向调节者 (facilitator) 注册登记。

```
Bind
: content tcp://medical: 8010
/* 采用 TCP 协议,端口为8010,主机为 medical
: sender user
: receiver tcp://facilitator, 1000
: in-reply-to ID1 /* 信息标志
: language Prolog
: ontology Med /* Med 是有关医学处理术语的集合
```

2. 用户初始化,接收目标,向网络中的调节者 (facilitator) 发送请求 Query,获得能完成处理目标的 agent A (Query-Result),agent A 评估自己的能力,决定是否接受处

理目标,并将决定结果返回 facilitator,若接受任务,agent A 从自身知识库中得到图像划分算法,将图像进行划分,开辟共享空间,并定好任务处理表 agent A 向 facilitator 发送请求寻找协作伙伴。

Query

```
: content image segmentation
: sender user
: receiver tcp://facilitator: 1000
: in-reply-to ID2
```

Query-Result

```
: content Agent A: image segmentation,
: sender server /* facilitator 名字
: receiver tcp://medical: 8010
: reply-with ID2
```

3. facilitator 通过注册表寻找到协作伙伴并将相应的 URL 反馈给 agent A (相应原语限于篇幅省略),agent A 根据 URL 将任务进行分配,如 agent B 进行区域分割,agent C 进行表面分割等,各协作代理准备就绪则开始协作处理。相应参数和结果及任务流程可以在共享空间中进行交换和变更。

4. 各代理进行协作处理,不断查询分割进展,传递相关的信息,其间可终止某一代理的工作,再选择其他代理进行。等各个代理处理完毕后,agent A 将信息集成,将结果返回用户,并进行指标评定,退出。不满意则重新开始分割进程。

结论 医学图像协调处理是为了解决当前医学影像处理系统中计算量大,时间效率低,图像质量差等客观问题而提出的。本文将 KQML 语言和多 agent 协作机制引入到协调处理中,能较好地解决协调过程中的信息交换与知识处理等问题,并且在实际应用中,KQML 语言建立在 CORBA 底层通信机制上,有机地和 CORBA 结合,KQML 提供了标准的信息交换格式,为各 agent 之间信息传递提供了有力支持。采用 KQML 和 agent 机制的医学图像协调系统具有开放性并能适应新的应用环境,尤其是 Web 应用环境。

参考文献

- Huang J, Jennings N R, Fox J. Cooperaton Distributed Medical Care. In: Proc. Second Int. Conf. on Cooperative Information Systems (CoopIS-94), Toronto, Canada, 1994. 255~263
- Lanzola G, et al. a framework for building cooperative software agents in medical applications. Artificial Intelligence in Medical, 1999, 16: 223~249
- Horsch A, et al. Collaborative Work with Medical Images in a University Hospital Environment, Proceedings JENC7
- Becker E, et al. a system for cooperative work in the medical domain. <http://www.gris.uni-tuebingen.de/>.
- Finin T, Fritzson R. KQML as an agent communication language, the Proceedings of the third international conference on information and knowledge management (CIKM'94), ACM Press, Nov. 1994
- 孙宪鹏,张宇,等.基于 KQML 语言的合同网协议模型及实现.信息与控制,2000,29(5):454~460
- 1998
- Padhye J. Towards a Comprehensive Congestion Control Framework for Continuous Media Flows in Best Effort Networks. A Dissertation of J. Padhye. March 2000
- Wu Wenjun, Xu Ying, Lu Jian. SRM-TFRC: a TCP-friendly multicast congestion control scheme based on SRM². ICCN-MC2001
- Handley M, Padhye J, Floyd S, Widmer J. TCP Friendly Rate Control (TFRC): Protocol Specification. Internet draft draft-ietf-tsvwg-tfrc-02.txt, work in progress, May 2001
- Stocia I, et al. A Hierarchical Fair Service Curve Algorithm for Link-Sharing, Real-Time and Priority Services. IEEE/ACM Transactions on Networking, 1997, 9
- Chawathe Y, et al. RMX: Reliable Multicast for Heterogeneous Networks. INFORCOM 2000, Tel Aviv, Israel, March 2000

参考文献

- Floyd S, et al. A Reliable Multicast Framework for Light-Weight Sessions and Application Level Framing. In: Proc. of ACM Sigcomm'95. 342~356
- Floyd S, et al. Equation-Based Congestion Control for Unicast Applications: the Extended Version. Feb 2000. <http://www.aciri.org/tfrc/>
- Handley M, Floyd S. Strawman Specification for Reliable Multicast Congestion Control. <http://north.east.isi.edu/~mjh/rmcc.ps.gz>
- Padhye J, et al. Modeling TCP throughput: a simple model and its empirical validation. ACM SIGCOMM'98, Vancouver, CA, Sep.

(上接第36页)