

基于 Internet 信息采集和决策的 Multi-Agent 系统的设计

袁平 张伟

(重庆教育学院计算机与现代教育技术系 重庆400067)

摘要 Internet 数据的特点,阻碍了人们充分有效地利用 Internet 信息。传统的集中式方法侧重于对长期积累数据的趋势分析,无法适应动态、实时的信息变化,因此采用 Multi-Agent 技术是利用 Internet 数据进行决策支持的必然。本文构建了一个用于多个异构数据源监测、集成、分析并进行决策的 Multi-Agent 系统,文章重点描述了该系统的组织情况,提出了在 Agent 体系结构基础上开发可重用部件的思想,从而提高了 Agent 创建的效率。文中还对试验进行了说明,最后作了小结。

关键词 多 Agent 组织, TCA 体系结构, 可重用软部件, 任务树

A Multi-Agent System Design Based on Gathering and Deciding the Internet Information

YUAN Ping ZHANG Wei

(Department of Computer & Modern Education Technology, Chongqing Education College, Chongqing 400067)

Abstract The characteristic of Internet data has caused people difficult to obtain Internet information efficiently. The centralized approaches pay more attention to the sequential analysis based on the long-term accumulated data, it must be not suitable for the information changed constantly. So we make use of the multi-agent technology to monitor, integrate, analyze and make decisions the distributed, heterogeneous data sources in Internet. In this paper, the organization of the multi-agent system is mainly described, a viewpoint of designing the reusable software component based on the universal architecture is proposed in the meantime, so the efficiency of creating agent is improved. Finally, we give out the example to illustrate the correctness of the proposed approach and summarize the research.

Keywords Organization of the multi-agent system, Architecture of TCA (Task Control Architecture), Reusable software component, Task tree

1. 引言

Web 迅速发展,数据量急剧增加,其中蕴藏了无限商机。但是 Web 数据分布、异质、半结构和动态等特点,极大地阻碍了人们充分有效地利用这些数据进行决策。当前,利用 Web 数据进行决策支持,大多采用数据挖掘的思路。其大体过程是这样:首先将若干异构数据源集成到一个统一的数据仓库中,然后采用 OLAP 或者数据挖掘的方法获取有用知识,还可以用各种图表来表现所发现的知识。这些知识往往是对全局发展趋势的预测或表现,而知识的利用,最终还需要人类的领域专家来评价和解释^[1,2]。

利用 Web 数据辅助用户决策的另一种思路,是采用 Agent 的思想,即通过 Agent 不断监测和获取相关数据源中的数据,并根据人类预先设定的目标,为用户提出建议。这样的系统很多,比如 Letizia^[3]、WebWatcher^[4]等,它们均已经实现,并可以应用。单一的信息 Agent 是比较容易实现的,因为它们所监测的数据源相对较少,但却难以满足 Internet 信息分布动态的特性。比如,对于 Internet 中存在的大量金融站点,这些站点的金融信息实时性很强,更新迅速,需要随时监测。很难利用一个单一的 Agent,或者一个数据仓库系统等集中式的技术完成对这类数据的分析。然而基于分布人工智能的 Multi-Agent 技术,却能够很好地应付这样的问题。文[5]介绍了一个用于金融证券管理的多 Agent 组织,可以监测多个异构数据源的金融数据,可以对规则库不断更新,但是其学习算法过于简单,难以应付金融中大量的领域知识。文[6]介

绍了一个用于学术会议安排的多 Agent 组织,其对任务 Agent 的设计过于简单,难于完成复杂的工作。

本文设计了一个基于 Multi-Agent 的组织,它能够访问 Internet 上的异构数据,能够不断监控和查询这些数据源,对它们进行集成和分析。Multi-Agent 技术设计的数据监控分析系统,更适应 Internet 中数据的特点。但在 Multi-Agent 系统中,必然要涉及到多个 Agent 的设计和实现,涉及到 Agent 之间的通信,涉及到 Agent 之间的冲突解决策略等,因此 Multi-Agent 技术相比过去集中式的方法而言更为复杂。

本文讨论的重点在于:1)系统的组织情况,即如何将这样一个分布的、松散的系统组织起来,完成用户提交的查询请求;2)如何能够快速高效地构造系统中的每个 Agent,并使得它们具有统一的界面,即采用了 TCA 体系结构;3)文中对实验情况进行了介绍,最后作了小结。论文的创新之处在于,提出了在统一体系结构基础上开发可重用软件部件的思想,从而提高了 Agent 创建的效率。

2. Agent 的组织、协调与交互

图1是分布式的 Agent 组织和交互框图。由图可见,系统中有3类 Agent:界面 Agent、任务 Agent 和信息 Agent。界面 Agent 与用户进行交互,获得用户提交的查询请求,并向用户提交最终的查询结果,同时还可以为用户建立个性化的用户模型,以此让系统与用户协作起来,完成任务,比如,界面 Agent 可以根据用户的兴趣过滤电子邮件。任务 Agent 将问题求解的任务形式化,并通过查询或与其它 Agent 进行信息

交换,来执行这些任务,例如,在一个用于证券管理的系统中,进行股票出售或买进的推荐,就是任务 Agent 的工作。信息 Agent 根据查询请求访问各种异构信息源,也可在数据源中监控是否有满足特定条件的信息出现,另外,在信息 Agent 中还保存有信息源的模型、信息访问策略等知识。

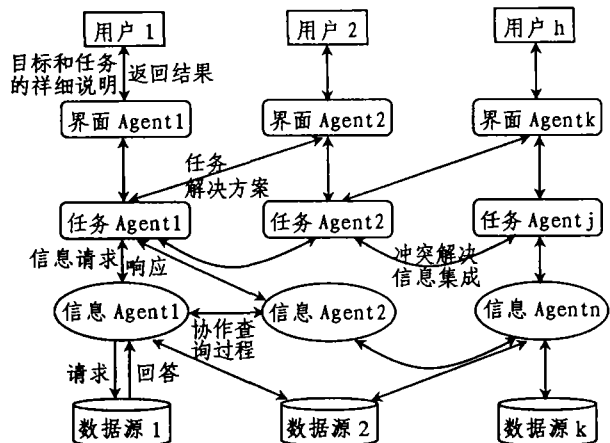


图1 系统的组织结构

该系统采用自顶向下的方式激活相关 Agent,并形成一个有组织的 Agent 团队。在顶层,界面 Agent 获取用户提交的查询请求,并将其转换为内部表示的任务规格说明——任务树,并将它传递给任务 Agent。这些任务树说明了任务/子任务的树状层次关系,任务 Agent 能够根据任务树,安排这些任务之间执行的先后顺序关系(第四部分有详细描述)。任务的执行最终依赖于信息 Agent,查询工作最后由信息 Agent 来完成具体的工作。信息 Agent 可以完成的查询有三类:1)相关数据源的一次查询;2)相关数据源的定期查询(例如,每隔30分钟查询 IBM 的股票价格);3)根据查询条件而监测数据源的变化(例如,查询是否在某事发生后,IBM 的股票价格在15分钟内低于 \$80)。很有可能多个信息 Agent 相互协作,共同完成一个查询目标,其结果提交给任务 Agent 之后,由任务 Agent 将多个查询结果进行合并、过滤。任务 Agent 在完成各自的任务时,会遇到资源限制等而相互冲突的情况,任务 Agent 之间可以相互协商。在这样一个多 Agent 的组织中,Agent 之间的通信主要采用 KQML 语言。Agent 之间的协商可以参见文[4],由于篇幅有限,在此不再详述。

3. Agent 的体系结构与 Agent 的高效构造

从软件工程的角度来看,如果能够开发出具有相当抽象程度,不依赖于具体应用领域的可复用软件部件,则能够加快软件开发进度,节约开发过程中资金和劳力。在我们的 Multi-Agent 系统中,也采用了软件工程的思想快速构建 Agent,即在统一的体系结构之下,开发可复用的软部件。如图2所示,本系统中的各类 Agent 都遵从 TCA(Task Control Architecture)体系结构。其中的各个软件模块,在一定参数带入后,可以实例化成为适合于特定应用环境的模块。

TCA 体系结构的核心在于任务树。任务树,是以“树”的数据结构来表示任务/子任务之间的层次关系,将等待继续分解的“任务”,即还没有落实到具体执行步骤的“任务”,作为树的非终结点;而将分解到最底层,即可以直接执行的活动,作为叶子结点。各个结点之间还可以加入时间约束,从而可以约束任务之间、执行行为之间、任务与执行行为之间的顺序关

系。正是这种任务的层状分解与时间约束的结合,形成了 Agent 内部的“任务”表示机制。

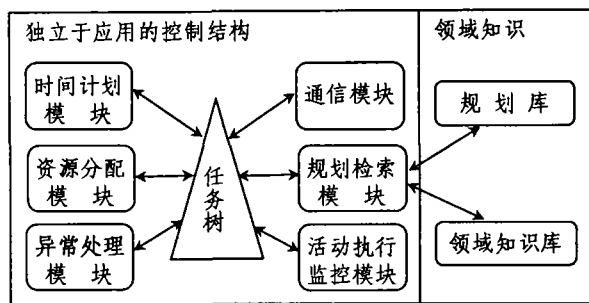


图2 Agent 体系结构—TCA

任务树的生成则是通信模块、规划检索模块、规划库协调工作的结果。

通信模块扩展了原有的 TCA 结构。它的主要功能在于,通过 KQML 语言,实现 Agent 之间的消息传递。Agent 所获取的查询目标(任务),正是通过该模块得到的,并将此目标传递给规划检索模块。规划检索模块,将从规划库中检索出合适的案例,指导任务树的生成。

规划库是采用“案例推理”的学习方式,即规划库最初以几条简单的案例开始,在每一次信息采集的任务完成之后,信息采集成功和失败的任务规划片断将存储下来。在每一次信息采集的任务完成之后,规划库都将被更新。“案例推理”方式的好处在于:1)最初的规划库可以由较简单的几个案例组成,随着 Agent 不断与外界交互,从而获得不断更新的知识,丰富规划库;2)每一次信息采集任务的完成,都将为下一个任务的规划提供经验,从而能对新任务进行有效规划,尽可能避免失败;3)规划库的不断更新,与问题求解同步进行,自动完成,无需额外编程。

时间安排模块,根据任务树中结点之间层次关系和时间约束关系,安排各个结点的执行顺序。该模块的输入是 Agent 当前处理任务的树以及其中的时间约束,而输出则是即将执行的活动。

行动的最终执行由“活动执行监控模块”和“异常处理模块”来完成。活动执行监控模块将 Agent 下一步将要执行的活动作为输入,然后准备该活动执行的各种环境,这其中包括各种资源的使用限制。在执行的过程中,不断监视各种资源使用的情况,如果资源的使用超过了定义的界限,则活动的执行将被中断,并标记该活动执行失败;否则,顺利完成活动的执行。标记为失败的活动,将在异常处理模块中对失败点重新规划它的执行方案。比如,若 Agent 在等待某外部资源数据时超时,将标记该活动失败;这时,异常处理模块会建议 Agent 搜索其它信息源,或者先处理其他任务。

领域知识库中存储了关于 Agent 功能(界面、任务、信息)的基本知识。如果是界面 Agent,需要包含关于用户兴趣爱好方面的知识。如果是信息 Agent,则其中会含有一个本地信息库,其中存储了 Agent 正在监测的外部数据源中的相关记录。

在 TCA 结构下各个模块都以抽象类的方式建立,可以根据具体应用的需要实例化,即可轻松生成界面 Agent、信息 Agent 和任务 Agent。具体而言,开发一个新的 Agent 可以简化成为如下步骤:

- 1)构建特定应用领域的知识库;

2) 构建应用于特定任务的规划库;

3) 用面向特定应用领域的规划库和知识库,对 TCA 体系结构实例化。

4. Agent 任务规划和执行的实例说明

我们以一个任务 Agent——个人信息查询器为例,说明 TCA 各个软件件的协调工作状况。个人信息查询器的主要功能在于:给出一个人的名字,找到与之相关的个人信息,如他的头衔、个人的电话号码、办公室电话号码等。我们的实验是基于国际学术会议的个人日程安排,因此信息 Agent 监控的主要是 Internet 上各大学的学术团体和学术成员数据库的数据。

首先,个人信息查询器直接从用户界面或其它 Agent 处,获得查询任务“采集某人员信息”。通信模块获取此消息,由于此任务无法直接执行,因此通信模块将该任务和相应参数(如该人员的姓名)提交给规划检索模块;规划检索模块检索规划库,找到合适的规划案例,并用参数对此案例进行实例化,这样将产生如下几个子任务或可执行的行动:“选择信息源”行动、“访问信息源”任务、“冲突解决”任务、“信息结果集成”任务。这样即可将这些行动和子任务添加到任务树中,如图3所示。任务树中作为叶子结点的“选择信息源”执行之后,再将“访问信息源”任务提交给通信模块,该模块又将此消息提交给规划检索模块,从而进一步将任务树向下生长。由此可见,任务树总是反映了任务以及任务执行的当前状况,且是不断更新。值得注意的是,在图3中,“采集某人员信息”的四个子任务之间存在先后顺序关系;而对于“访问信息源”的三个子任务,则可同时并发执行。

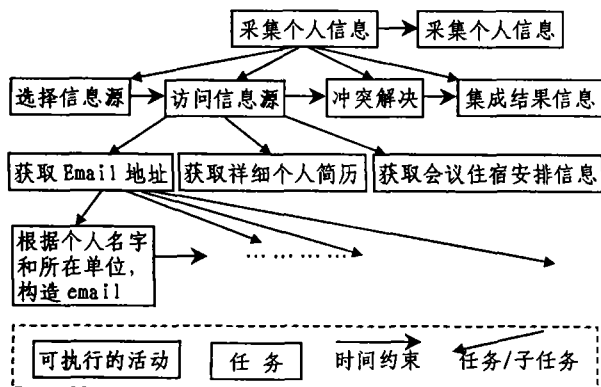


图3 个人信息查询器的任务树

5. 实验

我们进行的实验是,为到 CMU 大学进行会议交流的学者安排个人学术访问日程。以 CMU 大学为试验对象的主要原因在于,可以从网上获得 CMU 大学的学术团体、学术成员的详细数据^[6]。

为完成此任务,我们建立了一个由多 Agent 构成的系统,称为“访问学者招待系统”。它的主要功能在于:如果输入来访者的请求、希望在此大学停留的时间以及来访者的研究兴趣,则最终产生满足来访者学术兴趣的详细日程安排,包括出席时间、地点、参加人员的名字。

访问学者招待系统中,界面 Agent 主要用于在 Internet 上与访问者交互,获取他们的请求。任务 Agent 有三个:1)个

人信息查询器,如上所述,可以查询访问者个人的详细信息,而且还可以查询 CMU 学术社团的详细信息,以便二者可以很好匹配;2)来访者时间安排 Agent;3)学术团体日程管理 Agent。信息 Agent 较多,主要监视两类数据源:1)CMU 大学关于学术团体研究兴趣的数据库,由兴趣 Agent 监控;2)监控其他大学数据库的信息 Agent,以便从各个大学数据库中检索个人信息。

在实验中我们可以看到这样一个交互过程:

1) 来访者向系统提交请求,要求输入如下信息:名字、来访时间、逗留时间、研究兴趣等。

2) 系统提取来访者的名字、学术兴趣、所在单位。

3) 系统将来访者的学术兴趣提交给“兴趣 Agent”,兴趣 Agent 将在 CMU 中寻找兴趣类似的学术团体和成员。

4) 系统将来访者的名字和所在单位提交给“个人信息查询器”,它将发现关于来访者更多的详细信息,诸如在单位的职称、来访者正在参与的研究项目等。这部分信息之后将提交给“学术团体日程管理 Agent”,从而确定学术团体成员对此访问者的兴趣有多大。

5) “兴趣 Agent”搜索到相应的学术团体和成员后,将他们的名字提交给“个人信息查询器”。该 Agent 将在 CMU 大学的三个数据库中进行查询:Finger、CMU Who's-Who 和 CMU Room Database,从而发现关于学术团体更为详细的信息,甚至还需要解决信息二义性问题,并集成结果。

6) “兴趣 Agent”与“个人信息查询器”的合作可以获得与访问者兴趣一致的学术团体的集合,用户可以在此集合中进行选择。

7) 系统此时将自动向“学术团体日程管理 Agent”提交消息,询问是否愿意与该访问者会晤,以及会晤时间。

8) 系统采集这些意见,将其传递给“来访者时间安排 Agent”,然后确定来访者最终的时间安排。

小结 本文描述了采用重用软件件,构建一个用于监控、采集、集成多个分布异构数据源的多 Agent 系统。本文的重点在于该系统的整体组织,以及内部 Agent 的体系结构。对于多 Agent 之间的协商机制,由于篇幅所限没有过多描述,可以参考相关文献。本文用实例和试验说明了整个系统的多个 Agent 的组织过程,验证了设计的合理性。

参考文献

- 1 Garofalakis M N, et al. Data Mining and the Web: Past, Present and Future. In: Proc. of WIDM99, Kansas City, U. S. A., 1999
- 2 Third WEBKDD. Workshop on Data Mining for Web Applications: Mining Log Data Across All Customer TouchPoints. San Francisco, CA, USA, Aug. 2001
- 3 Decker K, Sycara K, Zeng D. Designing a Multi-Agent Portfolio Management System. In: Proc. of the AAAI Workshop on Internet Information Systems, 1996
- 4 Sycara K, Zeng Dajun. Towards an Intelligent Electronic Secretary. In: Proc. of the CIKM-94 (Intl. Conf. on Information and Knowledge Management) Workshop on Intelligent Information Agents, National Institute of Standards and Technology, Gaithersburg, Maryland, Dec. 1994
- 5 Simmons R. Structured Control for Autonomous Robots. IEEE Journal of Robotics and Automation, 1994
- 6 Sycara K, Zeng D. Multi-Agent Integration of Information Gathering and Decision Support. In: Proc. of ECAI-96