

一个维护知识库的软件 agent 互操作系统*

沈宁川 龙翔 聊鸿斌

(北京航空航天大学计算机系 北京 100083)

摘要 本文首先简介了基于 agent 的软件工程技术,然后利用这一技术,设计了一个用于知识库维护的软件 agent 互操作系统 SAIS (software agent interoperation system). SAIS 系统是由多问题求解器构成,每个问题求解器是一个软件 agent. 本文给出了 SAIS 的结构,并描述了 SAIS 的实现.

关键词 软件 agent, 互操作, 实现.

中图法分类号 TP311

用扩充逻辑程序设计语言作为工具建立知识库是知识信息处理最基本的应用之一.

随着需要解决的问题变得越来越复杂,知识库的规模也变得越来越,因此对知识库的协调性检查是十分必要的. 当知识库出现矛盾时,必须对知识库进行维护,以便保证其协调性. 为了有效地维护知识库,需要给用户提供一个实用工具.

设计和实现一个知识库维护的实用工具涉及到许多方面的内容,如用户界面、编辑器、知识库的协调性检查、知识库维护的操作、知识库的编译和运行等. 实际上,该工具是一个多问题求解系统,所以它应该包括维护知识库所涉及到的多问题求解器及求解器之间的互操作机制.

近年来,基于软件 agent 的软件工程技术^[1,2]为软件之间的互操作和软件集成提供了良好的模型. 正是利用这一技术,我们设计了一个维护知识库的软件 agent 互操作系统 SAIS (software agent interoperation system),它由多问题求解器构成,各问题求解器相互操作共同完成对知识库的维护.

在 SAIS 系统中,每个问题求解器都是一个软件 agent,并且可以分布在计算机网络的各个节点上. 它们可以在不同类型的主机、不同的操作系统上实现;并且相互操作. 因此,SAIS 系统与其它分布式系统一样面临着如支持异构环境下的互操作性、实现网络通信的透明性和资源(在 SAIS 系统中就是各个问题求解器)分布的透明性等问题. 本文针对这些问题,提出了我们的解决方案.

* 本文研究得到国家 863 高科技项目基金资助. 作者沈宁川,1964 年生,博士,副教授,主要研究领域为人工智能,软件工程. 龙翔,1963 年生,博士,副教授,主要研究领域为系统结构, VLSI. 聊鸿斌,1973 年生,硕士生,主要研究领域为软件工程,操作系统.

本文通讯联系人:沈宁川,北京 100083,北京航空航天大学计算机系

本文 1996-12-09 收到修改稿

1 基于 agent 的软件工程技术

1.1 agent 通信语言

标准的通信语言是克服异构环境的技术之一. 只要程序遵守这一规范, 就可以隐蔽异构环境造成的诸如数据结构、算法等等的细节差别, 从而实现异构环境下的软件互操作.

基于 agent 的软件工程技术通过定义一个广义通信语言来解决环境的异构问题.

由 ARPA 资助的 Knowledge Sharing Effort 研究小组设计的 agent 通信语言 (ACL) 是这种广义通信语言中的一种. 有关 ACL 的详细介绍见文献 [3, 4].

1.2 软件 agent

我们称一个可以正确使用 ACL 进行通信的软件实体为一个软件 agent. 这不仅意味着软件 agent 必须发送和接收 ACL 消息, 而且还意味着软件 agent 必须遵守隐含于那些消息中的行为约束.

典型的软件 agent 执行可分为两个阶段; 第 1 个阶段是初始化阶段, 完成内部数据结构初始化、注册等任务; 第 2 个阶段是一个消息循环, 不断地接收消息, 进行分析和处理.

1.3 联邦结构和 facilitator

有了可以相互通信的软件 agent, 就要选择一种体系结构来规范各个 agent 之间应该怎么通信. 通常有两种不同的方法来解决这一问题. 一种是直接通信, 即通信双方直接交谈. 另一种则是间接通信, 即通信双方由特殊的系统软件 agent (称为 facilitator) 来协调通信.

在软件 agent 互操作系统中, 通常由 facilitator 和其它 agent 构成联邦结构 (Federation Architecture). 图 1 是一个联邦结构的例子. 其中, 整个系统被划分为一个个的社区 (Community), 每个软件 agent 分别属于一个社区. 每个社区都有且仅有一个 facilitator. 在联邦结构中, 只有 facilitator 可以和其它社区的 facilitator 通信, 其它软件 agent 则只能同本社区的 facilitator 通信, 而 facilitator 则完成将消息发往目的地的任务. 事实上, facilitator 所提供的服务远不止如此, facilitator 存在真正目的是隐蔽资源的物理分布, 即每个软件 agent 无需知道通信的对方是谁, 只要其能提供对应的服务即可. 这将给分布环境下的软件集成带来诸多便利. 整个系统中多个 facilitator 相互操作, 向每一个软件 agent 提供了一个内容丰富的服务网.

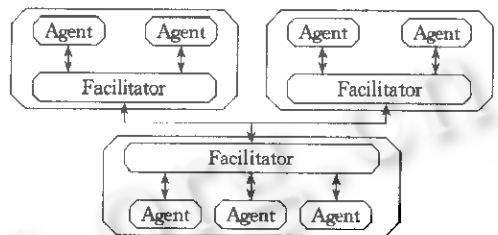


图1 联邦结构

tor 通信, 其它软件 agent 则只能同本社区的 facilitator 通信, 而 facilitator 则完成将消息发往目的地的任务. 事实上, facilitator 所提供的服务远不止如此, facilitator 存在真正目的是隐蔽资源的物理分布, 即每个软件 agent 无需知道通信的对方是谁, 只要其能提供对应的服务即可. 这将给分布环境下的软件集成带来诸多便利. 整个系统中多个 facilitator 相互操作, 向每一个软件 agent 提供了一个内容丰富的服务网.

2 SAIS 系统的构造

2.1 agent 的分类及其功能

SAIS 系统由多问题求解器构成, 每个问题求解器是一个软件 agent. SAIS 中的软件 agent 的分类及其功能如下:

- AUI 用户界面: 完成用户和 agent 之间的通信.
- ASE 综合编辑器: 编辑一个知识库并对其中的规则做语法检查.

- ACCE 静态协调性检查器:在编辑时对知识库的协调性检查.[5]
- AD 规约器:规约知识库的最佳修正.[6]
- AC 编译器:编译一个知识库.

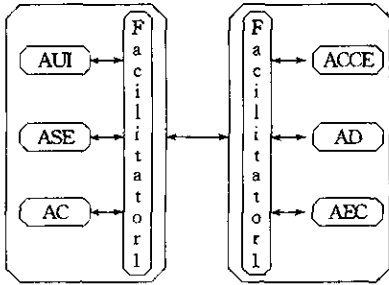


图2 SAIS系统的结构

- AEC 执行控制器:控制一个知识库的运行.

2.2 SAIS 系统的结构

图 2 所示的是 SAIS 系统的结构,其中有两台主机,形成两个社区.每个社区各有 1 个 facilitator,分别称为 facilitator1 和 facilitator2,其中一个社区有 3 个软件 agent,分别是 AUI,ASE 和 AC;另一个社区也有 3 个软件 agent,分别是 ACCE,AD 和 AEC. agent 只限与其局部的 facilitator 直接通信.

3 SAIS 系统的实现

3.1 SAIS 系统中的 API

SAIS 系统使用 Enterprise Integration Technologies 公司提供的 KAPI 作为它的 API (应用程序接口).

KAPI(KQML API)是为用 KQML 作为通信语言的软件 agent 提供的一组 C 语言库函数,下面是 KAPI 的一些基本概念:

• AGENT

AGENT 是一个用 KQML 作为通信语言可以同其它 AGENT 通信的程序,KAPI 为 AGENT 的开发提供了一个高级接口.

• KQML 消息

KQML 是 agent 之间交换信息的标准语言.KAPI 最小地实施对 KQML 标准的依附.

• AGENTNAME

AGENTNAME 对 agent 来说是一个唯一的标识,它是一个 ASCII 码的字符串. agent 通过使用 AGENTNAME 向其它 agent 发送消息.

• URL

对一个 agent(或其它网络资源)来说,URL 是一个传输地址.

• ANS

ANS 是一个用来查找 URL 及其特定的 AGENTNAME 的网络资源.

KAPI 提供 C 函数库支持 agent 通信.利用多种传输机制和协议 KAPI 使得 agent 发送和接收消息变得更容易.

3.2 软件 agent 的结构

软件 agent 的执行分为两个阶段:初始化阶段和消息循环阶段,典型的软件 agent 具有如下框架:

```
AGENT()
{
    Initialize();          /* 完成初始化动作 */
    ...
}
```

```
while (! done) {  
    GetMessage(&msg); /* 读一条消息 */  
    ParseMessage(&msg); /* 分析 */  
    Do(&msg); /* 数据分析结果执行相应动作 */  
}  
return (done);  
}
```

其中 Initialize() 完成如对 ANS 注册、向本社区 facilitator 发送兴趣文档等工作。对于不同的软件 agent, 由于其实现的功能与应用领域相关, 因此仅给出了一般的 agent 框架, 具体的功能则因具体领域而异。

3.3 软件 agent 名字服务器

软件 agent 名字服务器(ANS)是实现网络通信透明性的技术之一。

在 SAIS 系统中, ANS 完成从 AGENTNAME 到 URL 的映射, 从而使软件 agent 之间仅用 AGENTNAME 来标识通信的对方。与一般的名字服务器不同(如 Internet 中的 Domain Name System), ANS 的映射是动态联编的(Dynamic Binding), 即与 AGENTNAME 对应的 URL 是变化的, 而且可能变化频繁。这样做的一个目的就是更好地体现软件 agent 的通信透明性。为此要求每个软件 agent 在初始化阶段从系统分配到一个 URL 后, 再向 ANS 登记注册, 标识自己的存在。而当软件 agent 从一台主机移到另一台主机时, 则可以先注销, 然后再向 ANS 登记。由此 ANS 始终保持 AGENTNAME 到最新的 URL 的映射, 同时 ANS 还负责在软件 agent 的 URL 无效时, 对引用它的软件 agent 的通知。由于 ANS 的作用, SAIS 系统中任何通信都仅使用 AGENTNAME, 从而隐藏了通信细节, 提高了透明性。

3.4 SAIS 中的 facilitator

在 SAIS 系统中 facilitator 是一种特殊的软件 agent。它具有一般的 AGENT 所具有的一切基本特征, 同时完成协调其它软件 agent 之间通信的管理。软件 agent 之间的通信完全由本社区的 facilitator 和其它社区的 facilitator 协调完成。在 SAIS 系统中任意两个 facilitator 之间可以通信, 同时选出一个 facilitator 作为监控器(Monitor)负责完成 facilitator 的建立和退出。

4 结 论

利用基于软件 agent 的软件工程技术是解决大型复杂的互操作问题的有效途径。SAIS 系统正是利用这一技术的尝试。这一技术较好地解决了分布计算中的诸多问题。我们相信随着 agent 技术的不断发展, 它将成为分布计算的主流技术之一。

参 考 文 献

- 1 Genesereth M R, Ketchpel S P. Software agents. Communication of the ACM, 1994, 37(7): 48~53.
- 2 Genesereth M R, Singh N, Syed M. A distributed anonymous knowledge sharing approach to software interoperation. In: Proc. of the International Symposium on Fifth Generation Computing Systems, 1994. 125~139.
- 3 Genesereth M R, Fikes R E. Knowledge interchange format version3 reference manual. Logic-92-1, Stanford Uni-

- versity Logic Group, 1992.
- 4 External interface working group ARPA knowledge sharing initiative, specification of the KQML agent-communication language. Working paper, 1992.
 - 5 Shen Ningchuan, Long Xiang, Li Wei. Consistency checking and constraint condition in knowledge base maintenance. In: Proc. of 4th International Conference on Young Computer Scientists, 1995. 548~554.
 - 6 Li Wei, Shen Ningchuan, Wang J. R-calculus: a logical approach for knowledge base maintenance. International Journal of Artificial Intelligence Tools, 1995,4(1~2):177~200.

A SOFTWARE AGENT INTEROPERATION SYSTEM FOR MAINTAINING KNOWLEDGE BASES

SHEN Ningchuan LONG Xiang LIAO Hongbin

(Department of Computer Science Beijing University of Aeronautics and Astronautics Beijing 100083)

Abstract This paper briefly introduces the technology of agent-based software engineering, and then designs a SAIS (software agent interoperation system) for maintaining knowledge bases by using this technology. The SAIS system consists of multiple problem solvers, and each problem solver is a software agent. This paper also shows the structure of the SAIS system, and describes the technical aspects about the implementation of the SAIS system.

Key words Software agent, interoperation, implementation.

Class number TP311