

基于 Agent 技术的软件重用模式^{*}

Software Reuse Modes Based on Agent Technology

毛新军 王怀民 齐治昌

(国防科技大学计算机系602教研室 长沙410073)

Abstract Agent technology is a new software development technology, which can effectively support software reuse and will essentially change the software reuse manner. The paper analyzes how agent technology supports software reuse and its characteristics, presents two kinds of software reuse modes to reuse non-agent system and agent-based system respectively, in order to show how to reuse software based on agent technology and its validity.

Keywords Agent technology, Software reuse, Agent-based system

1 引言

软件重用是指利用已有的软件资源(如软件设计文档、源程序代码、可执行程序等)来进行软件系统的开发。有效的软件重用可以提高软件生产率和软件系统的质量,降低软件开发成本。因此,几乎所有的软件开发技术都提供了相应机制和设施来支持软件重用,如面向对象的软件开发技术、软构件技术等等。

软件重用面临着许多问题和挑战,例如如何开发可重用的软构件、如何发现可重用对象、如何进行软件重用等等,现阶段成功的软件重用项目并不多见^[1],原因是多方面的,从技术角度看,阻碍当前软件重用广泛、有效应用的一个重要因素是现阶段的软件重用方式仍停留于“静态软件重用”。

(1)重用对象的静态特征:重用对象通常是一些静态的软件资源如需求分析文档、设计文档、库函数、类库、软构件等,这些软件资源通常不能单独运作,软件开发人员需要要把它们与其它软件资源相组合、形成一个完整系统之后才能运行;

(2)重用对象缺乏灵活性:可重用对象通常具有预定义、多样式的接口,软件开发人员必须严格按照其提供的接口进行软件重用;同时可重用对象不能感知环境中的其它可重用对象,因而也就不能通过与其它重用对象的交互和合作来完成软件重用活动;

(3)软件开发人员在寻求重用:由于重用对象的静态性和缺乏灵活性,软件重用活动不得不由软件开发人员来完成。在软件开发过程中,软件开发人员根据软件项目的具体需求从已有的软件资源中寻找可重用对

象,并通过对它们的理解、修改和集成等工作来完成软件重用活动;

(4)重用对象的有限性:静态软件重用的可重用资源也比较有限,主要是软件开发人员和软件开发组织所掌握的软件资源。

Agent 技术是近年来出现的一项新颖的软件开发技术。与现有技术相比较,Agent 技术能够更为有效地支持软件重用,在一定程度上促进上述问题的解决,并将从根本上改变现有的软件重用方式。

2 Agent 技术及其对软件重用的支持

Agent 是目前计算机科学领域中的一个重要概念,它是指驻留于某一环境下,具备自适应性、社会性、能够持续自主地执行动作以满足系统设计目标的计算实体^[2,3]。Agent 技术强调基于 Agent 这一抽象概念模型(不同于对象)来构造和开发应用系统:系统由一个或者多个 Agent 构成(我们称这样的系统为基于 Agent 系统);每个 Agent 都是自主或者半自主的计算实体;系统的各个 Agent 间通过复杂的社会性交互来进行问题求解。因此,Agent 技术不同于现有的其它软件开发技术,它体现了一种新的软件开发方式用于概念化、分析、描述和实现应用系统。Agent 技术能够有效地支持软件重用。具体表现在:

• Agent 的可重用性更好:Agent 封装了状态和行为,并且能够对其内部状态和行为实施自主的控制(这区别于对象,对象仅能控制自身的状态),因此 Agent 的独立性更强;从软件工程的角度上看,其模块性、可维护性、可重用性也就更好。

^{*} 本文得到了“973”项目的支持。毛新军 博士,主要研究领域:软件重用,Agent 理论、技术和应用。

•通过灵活的社会性交互相互重用:在基于 Agent 系统中,资源、能力、数据和信息等分布在多个不同的 Agent 之中,Agent 之间通过灵活的社会性交互(协同、合作、协商等)相互重用资源、能力等,从而进一步地求解问题,因此与其它软件开发技术相比较,Agent 之间不是通过预定义、静态、多样化的调用接口(例如函数和方法调用)来实现软件重用,Agent 之间的交互接口非常单一,例如就是基于 ACL(Agent Communication Language)的消息传输;而交互内容则更加丰富,例如 ACL 的强大表达能力。

•软件重用的自适应性和动态性:Agent 驻留在一定的环境中(例如 Internet/Intranet),它能感知环境中的其它 Agent,以及这些 Agent 所能提供的可重用资源和能力,及时、自主地对其它 Agent 的社会性交互作出响应并为其它 Agent 提供可重用的资源和服务。在基于 Agent 系统中,Agent 对外提供的服务和资源、重用哪个 Agent 的资源和服务、如何进行重用等都是事先不能确定的,是动态的。

Agent 技术对软件重用的上述支持区别于现有的其它软件开发技术,它将能更加适应越来越多的开放、分布式应用系统对软件重用的需求。因此,如何利用 Agent 技术进行软件重用将是人们关注的一项重要研究内容。

3 基于 Agent 技术的软件重用

基于 Agent 技术进行软件重用的核心思想是:利用 Agent 这一抽象概念模型来刻画、描述、设计和构造可重用软件资源,通过 Agent 之间的社会性交互来进行软件重用。因此,基于 Agent 技术的软件重用涉及如何利用 Agent 技术构造可重用的软构件,如何利用这些软构件进行软件重用等问题的研究。前一问题一直是 Agent 技术的研究内容,已经取得了一定的成果^[2,4],针对这一问题本文将着重研究如何利用 Agent 技术对现有的大量非 Agent 软件进行改造,开发出可重用的软构件以对现有非 Agent 软件系统进行重用,保护已有的投资。针对后一问题,我们将研究如何基于 Agent 技术对基于 Agent 系统进行软件重用。基于 Agent 技术的这二种软件重用分别针对不同的对象,反应了 Agent 技术对软件重用支持的多样性,体现了基于 Agent 技术的二种不同软件重用模式。

3.1 对基于 Agent 系统的重用

对基于 Agent 系统的重用首先涉及到对 Agent 的管理。标准化的 Agent 管理设施可以有效地促进对基于 Agent 系统的管理以及在此基础上的软件重用。图1是用 UML 描述的、Agent 技术标准化组织 FIPA (Foundation for Intelligent Physical Agents)提出的

Agent 管理参考模型^[7]。该模型中的核心要素是 Agent 平台,用于 Agent 的配置和管理。Agent 必须在 Agent 平台上注册以便于与 Agent 平台内部或者平台间的 Agent 进行交互。Agent 平台主要包括以下几个部件:

•Agent 管理系统:管理对 Agent 平台的存取和使用;维护 Agent 的命名目录和相关的传输地址,为其它 Agent 提供“白页”服务;管理 Agent 的生命周期包括认证、注册、注销、修改、查询平台剖面、查找 Agent、控制 Agent 的生命周期;

•目录设施:管理 Agent 对外提供的服务,为 Agent 提供“黄页”服务,Agent 必须将其对外提供的服务在目录设施中注册,Agent 也可通过目录设施查询其它 Agent 提供的服务;

•安全管理:管理 Agent 平台的安全,包括审核、认证、加密等;

•Agent 通讯通道:为 Agent、Agent 平台、Agent 管理系统、Agent 服务之间的交互提供通讯通道。

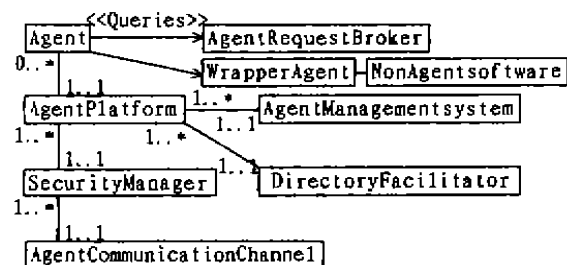


图1 Agent 管理参考模型

一个 Agent 要为其它 Agent 所重用,必须在 Agent 平台的 Agent 管理系统进行注册,并在 Agent 平台的目录设施中注册它对外提供的服务。只有在 Agent 平台注册的 Agent 才能为其它 Agent 所感知,才能与其它 Agent 通过 Agent 平台进行交互。Agent 对外提供的服务是动态的,以反应 Agent 的多面性和动态性。

为了有效地进行软件重用,一个基于 Agent 系统必须具备以下机制:(1)根据用户或者问题求解的需求,确定所需的资源和服务;(2)感知所在的 Agent 平台或其它 Agent 平台中的 Agent,以及它们对外提供的服务;(3)通过和相关 Agent 的交互和合作来重用它们的资源和服务。

要开发出一个功能全面、复杂庞大的基于 Agent 系统显然是不可取也是不可能的,因为用户的需求是动态变化的,而且这样的软件不利于管理和维护。因此,如何适应用户需求的动态变化是当前计算机科学工作者面临的一个重用的问题。Agent 技术强调开发

一组规模和粒度适中、易于管理和维护的可重用 Agent, 每个 Agent 具有有限的资源和能力, 致力于解决某个特定具体的问题。这些 Agent 之间通过交互合作、相互重用对方的资源和能力来共同求解更为复杂庞大的问题。这一思想一方面自然地反应了人类自身解决问题的模式, 另一方面与分布式人工智能的研究目标是相一致的^[4]。

这种基于 Agent 系统能够在一定程度上适应用户的动态需求, 并根据用户的需求通过诸如问题分解、规划制订等多种方式确定问题求解所需的资源和服务。对于具体的基于 Agent 系统而言, 问题分解和规划制订等机制可以是通用的, 适应一大类问题, 也可随用户需求的变化而动态变化。

Agent 可以通过 Agent 平台所提供的设施, 查询所在的 Agent 平台以及其它相关 Agent 平台上的 Agent 对外提供的服务, 确定哪些 Agent 可以为其提供所需的资源和服务以及这些 Agent 的详细信息。在此基础上, Agent 可以通过 Agent 通讯通道与相关 Agent 进行交互。Agent 之间的交互涉及交互语言、交互方式以及交互协议等方面的问题。目前, Agent 标准化组织如 FIPA 已就这些问题制订了相关的 Agent 技术标准, 例如 Agent 之间的交互可采用 KQML 语言, 或者 FIPA 提出的 ACL (Agent Communication Language) 语言; Agent 间可基于简单的消息传递进行通讯; Agent 之间的交互协议(或者称合作模型)可基于 FIPA 的相关标准以及文[5], [6]的工作。

3.2 对现有系统的重用

随着用户需求的不断提高及计算机科学技术的飞速发展, 如何有效地重用已有的大量遗留软件系统, 保护用户的投资成为目前许多用户和计算机工作者关注的问题。一方面, 用户需求的变化常常需要对已有的系统进行扩充和维护, 在软件系统的可维护性较差、缺乏相关文档的情况下开展上述工作显然是比较困难的, 而重新开发整个系统的代价很大, 不能有效地保护用户的投资; 另一方面, 目前已有大量的遗留软件系统, 它们提供了大量的功能以解决各种各样的应用问题, 而且许多系统经过大量、长时间的运行, 其可靠性、质量得到实践的检验。

Agent 技术可以促进对现有的大量软件系统的重用, 加强基于 Agent 系统与非 Agent 系统之间的互操作。我们可以基于 Agent 技术在遗留软件系统的基础上构筑一个 Agent (该 Agent 称为包装 Agent), 包装 Agent 的主要任务是确保基于 Agent 系统与非 Agent 系统及其服务之间的连接, 系统结构如图2所示。

从逻辑上看, 包装 Agent 在已有软件系统的基础上增加了 Agent 功能, 使得包装后的已有软件系统表

现出 Agent 的特征, 其它 Agent 可以通过 Agent 通讯通道与包装 Agent 进行灵活多样的交互合作以寻求对已有软件系统的重用, 包装 Agent 通过调用已有软件系统的接口或者激活已有软件系统相应的方法为其它 Agent 提供可重用的资源和服务, 并通过 Agent 通讯通道向其它 Agent 提供相应的结果。从物理上看, 包装 Agent 和已有软件系统可以作为二个独立的软件运行, 一个包装 Agent 可以管理一个或者多个已有的软件系统。



图2 对已有软件系统的重用

结论 Agent 技术是近年来计算机科学领域的一项新颖的软件开发技术, 它不仅能够促进分布、开发、动态系统的开发, 而且能有效地支持软件重用。本文分析了 Agent 技术对软件重用的支持, 提出了基于 Agent 技术的二种软件重用模式, 分别针对非 Agent 系统和基于 Agent 系统, 以研究如何利用 Agent 技术进行软件重用。

基于 Agent 技术的软件重用不同于现有的软件重用方式, 从本质上看它是一种动态的软件重用, 具体表现在: (1) 重用对象是高度自治、动态的软件系统, 它们拥有某些特定的资源, 实现了某种特定的功能, 这些可重用对象驻留于一定的环境之中, 它们能通过某种方式感知环境的其它可重用软件资源; (2) 重用对象表现出较强的灵活性, 它们之间不再通过对预定义接口的调用来进行软件重用, 而是通过社会性的交互合作来完成软件重用活动; (3) 软件重用活动由软件系统动态地完成, 通过感知环境来寻求能够满足其需求的可重用对象, 通过和可重用对象的交互合作来完成软件重用活动; (4) 重用的内容是开放的, 重用资源不再局限于一个或者一组函数库或者类库, 而是在局域环境内甚至是 Internet 和 Intranet 中的所有可利用软件资源^[9]。

基于 Agent 技术的软件重用引起了人们的关注和重视, 它可能对今后的软件开发产生重大而深远的影响。由于基于 Agent 技术软件重用的本质是通过自主 Agent 之间的交互和互操作来重用相关 Agent 的资源和服务, 而标准化是确保基于 Agent 系统互操作的一种重要举措, 因此许多 Agent 技术标准化组织如

(下转第121页)

位置都随机产生。图1为8个目标,8个量测的情况下的关联结果,图2为8个目标,10个量测情况下的关联结果,图中矩形表示目标位置,圆形表示量测位置,目标位置和量测位置间连线表示二者关联,未连线即孤立存在者为虚警。

结论 由以上实验结果可知,用遗传算法求解多目标数据关联问题是可行的,本文提出的算法关联成功率较高,具有一定的应用前景,值得继续研究。但如何提高算法的收敛速度,以满足目标跟踪系统的实时性要求,是需要进一步研究的关键。

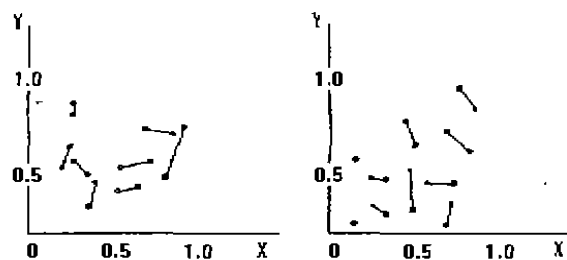


图1 目标、量测关联结果1

图2 目标、量测关联结果2

参考文献

- 1 Sittler R W. An optimal data association problem in surveillance theory. *IEEE Military Electronics*, 1964, 8 (2):125~139
- 2 Smith P, Buechler G. A branching algorithm for discriminating and tracking multiple objects. *IEEE Transactions on Automatic Control*, 1975, 20(2):101~104
- 3 Singer R A, Sea R G. New results in optimizing surveillance system tracking and data correlation performance in dense multitarget environments. *IEEE Transactions on Automatic Control*, 1973, 18(6):571~582
- 4 Morefield C L. Application of 0-1 interger programming to multitarget tracking problems. *IEEE Transactions on Automatic Control*, 1977, 22(3):302~312
- 5 Reid D B. An algorithm for tracking multiple targets. *IEEE Transactions on Automatic Control*, 1979, 24(6):843~854
- 6 Bar-Shalom Y, Tse E. Tracking in a cluttered environment with probabilistic data association. *Automatica*, 1975, 11 (5):451~460
- 7 Bar-Shalom Y. Extension of the probabilistic data association filter in multi-target tracking. In: *Proc. of the 5th Symp on Nonlinear Estimation*. 1974. 16~21
- 8 敬忠良. 神经网络跟踪理论及应用. 北京:国防工业出版社, 1995
- 9 Lazoff H. Target Tracking Using Fuzzy Logic Association, Acoustics, Speech and Signal Processing, 1998. In: *Proc. of the 1998 IEEE Intl Conf 1998*. 2457~2460
- 10 王宁, 郭立. 基于遗传算法的多传感器多目标静态数据关联. *火控雷达技术*, 1999(1):7~11

(上接第92页)

OMG 的 Agent 工作组、FIPA 等非常重视 Agent 技术标准化的制订工作以推动基于 Agent 技术的软件重用^[7]。

参考文献

- 1 Fraser S, et al. Software Reuse: Nemesis or Nirvana. *ACM SIGPLAN NOTES*, 1998, 33(10):417~420
- 2 Wooldridge M, Jennings N R. Intelligent Agents: Theory and Practice. *Knowledge Engineering Review*, 1995, 10 (3):115~152
- 3 Jennings N R, Wooldridge M. Application of Intelligent Agents. *Agent Technology: Foundation, Applications, and Markets 1998*
- 4 Wooldridge M. Agent-based Software Engineering. *IEEE Proceedings on Software Engineering*, 1997, 144(1):26~37
- 5 毛新军, 王怀民, 陈火旺. 一个主动的分布协同计算模型. *计算机学报*, 1997, 20(并行与分布增刊):147~152
- 6 毛新军, 王怀民, 陈火旺. Multi-agent 系统中的非终止主动合作计算模型. *计算机研究与发展*, 1999(9)
- 7 OMG Agent Working Group. Agent Technology Green Paper. 2000.4
- 8 O'Hare G M P, Jennings N R. *Foundation of Distributed Artificial Intelligence*, A Wiley-Interscience Publication, 1996
- 9 Mao Xinjun, Qi Zhichang. Software Reuse: From Object to Agent. *TOOLS Asia'99*, 1999. 10