移动 Agent 的影子协议

The Shadow Protocol of Mobile Agent

聂亚杰　刘大昕

（哈尔滨工程大学计算机与信息科学系　哈尔滨150001）

Abstract　This paper firstly introduces some mobile agent terminology, and then, brings forward the control algorithms of Mobile Agent Shadow Protocol.

Keywords　Agent, Mobile Agent, Shadow protocol, Dependency object

移动 Agent 控制协议是当前移动 Agent 技术研究的重点问题之一。影子协议可以用来定位 Agent、终止 Agent 和进行孤儿的探测。

1　影子概念

提出影子概念之前，需要了解一下能量和路径的概念。能量的概念是针对探测孤儿而提出的，是指 Agent 获得资源和接受服务的能力。在启动 Agent 时赋予它一定数量的能量。Agent 在生命期里，消耗它所在场所的资源（例如 CPU 时间、内存）、使用场所提供的服务（例如目录服务）。存取任何资源都要消耗 Agent 的能量，也就是说 Agent 存取资源的数量是有限的。

路径的概念是针对 Agent 的定位和终止而提出的。路径上的单个元素，即节点上的数据结构，被称为 Proxy。移动 Agent 通常以不可预见的方式移动，通常不能预测某时刻它所在的位置。但是如果 Agent 在迁徙之前把目标的位置信息留在老机器上，即留下一个 Proxy，于是一个个 Proxy 就组成了路径。创建 Agent 的场所称为锚场所（Anchor Place）。如果知道了锚场所，就可以循序找到路径。沿着这条路径最终会找到当前 Agent 的驻留位置。

影子的概念是能量概念和路径概念的融合，兼有两者的优点。每个应用程序可以在一个场所创建一个或多个称为影子的依赖对象。影子创建的场所不必要在运行应用程序的那台主机上。应用程序依赖影子创建 Agent，而不是应用程序本身，见图1所示。Agent 被应用程序启动后，就不再与应用程序联系了。应用程序可以在非永久连接网络的系统中间歇地执行。影子是一个依赖对象。如果 Agent 成了孤儿，它的影子就不存在了，反之亦然。

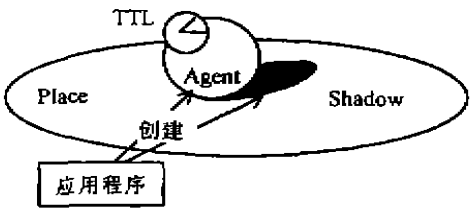

图1　创建影子

2　基本影子协议 b-SP

设定固定的时间间隔 TTL（Time To Live），检查每个 Agent 的相关影子是否存在。Agent 在进入检查期后就不允许迁徙了。如果因为节点崩溃导致场所不可达到，那么就会出现影子丢失，则 Agent 成为孤儿，可以被系统移除。如果场所不可达到的原因是网络的部分通信故障，那么影子可以正确地判断 Agent 已经终止，就移除 Agent 的依赖关系。如果 Agent 创建一个新的 Agent，则系统将原 Agent 的影子和剩余的 TTL 赋予这个新 Agent。这一点很重要，被创建的 Agent 只得到原 Agent 的剩余的 TTL，而不是全部。如果赋予全部的 TTL，则如果新的 Agent 再生成新的 Agent……，TTL 永远不会减少为0，Agent 不会终止。Agent 创建新 Agent，如图2所示。

当 Agent 到达一个场所时，如果它的 Proxy 不存在的话，就创建一个 Proxy。Proxy 包含 Agent 的 TTL、Agent 的 ID、影子的 ID 等信息。当 Agent 离开场所时，需要检查 Agent 的 TTL 是否大于0，决定它是否可以迁徙。如果 TTL 不大于0，那么就抛出一个异常。这样就避免了 Agent 在检查期迁徙。如果 TTL 大于0，就可以迁徙，则 Agent 的 Proxy 存储目标场所的

聂亚杰　工程师，硕士生，研究方向为数据库与知识库，刘大昕　教授、博导，研究方向为数据库与知识库、Internet/Intranet 应用软件、计算机图像处理。

位置信息，创建新的路径片段，另外还要根据 Agent 剩余的 TTL 值创建一个称为 Proxy-Time-Out 的计时器，用来判断这个 Proxy 是否可以被系统从路径中移除。Agent 驻留的场所定时地减少 Agent 的 TTL，一旦 TTL 减少到0，就返回给影子驻留的场所一个消息。这个消息包括 Agent 的 ID 和影子的 ID。同时启动 Check-Phase-Time-Out 计时器，Agent 进入检查期。

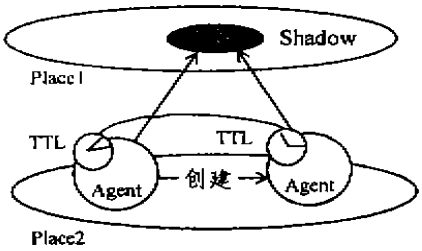

图2 Agent 创建新 Agent

检查消息由创建影子的场所接收。首先，影子停止 TTL-Time-Out 计时器。这个计时器用于探测已经终止的 Agent。影子决定 Agent 的新 TTL。一旦检查消息到达 Agent 驻留的场所，Check-Phase-Time-Out 计时器就停止了，Check-Phase-Time-Out 计时器、Agent 的 TTL 以及相应的 Proxy 的 TTL 被重置。检查期就结束了，Agent 被允许迁徙。

通过路径信息的帮助可以找到 Agent 的 Proxy。如果要查找的 Agent 就在本机，那么可以从本地的 Agent 列表中查到。如果不在本地，则通过 Proxy 指示的信息，给目标场所发送"Find"消息。如果在本地找不到相应的 Proxy，则返回一个错误消息。目标场所收到"Find"消息后，检查 Agent 是否在当地，如果在，返回一个成功的消息；如果不在，继续到下一个场所去找，直到找到 Agent 的场所，即找到路径的终点。

3 层次影子协议 h-SP

h-SP 和 b-SP 最主要的区别在于：在 h-SP 中应用程序影子不再知道所有依赖它的 Agent，而只知道具有直接依赖关系的影子。

当用移动 Agent 应用程序解决复杂问题时，往往将问题分解为小问题。每个小问题单独由不同的子 Agent 群解决。层次影子协议就是针对这个问题提出的。一个层次影子依赖于另一个影子而存在。层次影子为它的依赖对象所有，也就是说，Agent 可以被它的所有者移除，见图3所示。

如果一个拥有层次影子的 Agent 创建子 Agent，那么子 Agent 与层次影子关联，组成树形结构信息。也就是说，所有子 Agent 都可以一起被关联的层次影子的所有者(Agent)根据需要终止操作。

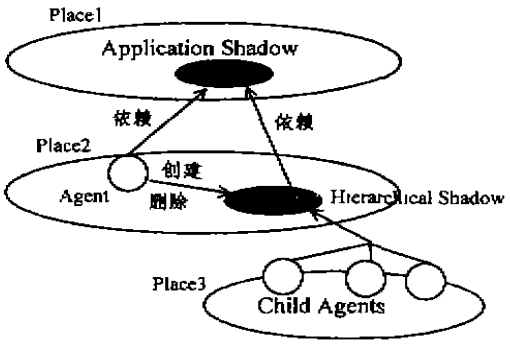

图3 层次影子

在层次影子协议模型里，场所只定时检查驻留在此场所上的层次影子，而不是所有的 Agent。

检查消息由层次影子的父影子所在的场所接收。找到父影子，层次影子关联的计时器被停止。父影子赋予子影子新的 TTL，然后计时器以新的 TTL 执行。于是影子的检查期结束。算法 h-SP2给出检查期的各个方法。

4 移动影子协议 m-SP

移动影子协议的根本思想是：影子驻留在某个场所时，影子与依赖它的 Agent 交互；影子离开时，仅留下影子的 Proxy；当 Agent 继续前进时，影子紧随其后，见图4所示。

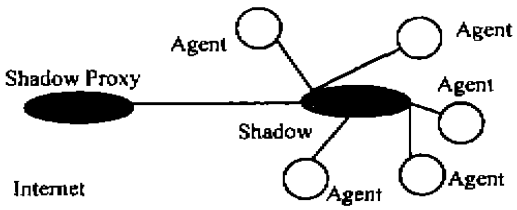

图4 移动影子

移动影子的操作包括从源场所发送到目标场所一个影子、启动一个计时器，计时器指示影子的 Proxy 何时被移除。算法 m-SP 给出移动和通信的各个方法。

算法 m-SP：

```
1) move(target)
 if (shadowTTL != 0)
 {
  send(target, ("MobileShadow",this));
  if (currentPlace != Null)
  {
   //shadow TTL + agent TTL + Timeout
   pathTimeOut = shadowTTl + ttl +timeout;
   starttmer(pathTimeOut,shadow);
   shadowTTL =-1;
  }
  currentPlace = target;
 }
```

（下转第67页）

解决 Web 个性化信息服务问题，在个性化技术的研究中作了新的探索。由于该领域的研究国内外都处于探索阶段，因此，我们认为，CMR 方法与 SmartWeb 系统在以下几方面有所创新：

·支持 Internet 环境中零输入个性化技术的、集数据仓库技术、数据挖掘技术、规则解析技术和信息集成技术于一体的体系结构。

·支持个性化需求的数据挖掘、Web 挖掘技术，包括数据源的存储结构及在该存储结构下的挖掘算法。

·Web 挖掘规则解析机制以及结构化存储模式转换机制。

·具有个性化特征的基于 XML 的信息集成和裁剪技术。

参 考 文 献

1 http://www.personalizaton.org/
2 Doug R Introduction:Personalized Views of Personalization. ACM Computer,2000,43(8):26～28
3 Manber U,Patel A,Robision J. Experience with Personalization on Yahoo! ACM Computer,2000,43(8):41～43
4 Malio P,Barrett R. Intermediaries Personalize Information Streams ACM Computer,2000,43(8):96～101
5 O'Leary D The Internet,intraners,and rhe AI renaisance lEEE Compuuer,Jan. 1997
6 Aggarwal C,Yu P. Data Mining Techniques for Personalization. lEEE Data Engineering Bulletin,2000,23(1):4～9
7 Ceri S,Fraternali P,Paraboschi S. One-ro-One Personalization of Data-Intensive Web Sites. In:Proc. of VLDB'2000
8 Joachims T,et al. WebWatcher:A Tour Guide for the World Wide Web. In,Proc. of the Int. Joint. Conf in AI(IJCAI97),Aug. 1997
9 Nicholas K,McKee J,Toolan F. Towards Zero-input Personalization:Referrer-Based Page Prediction. In:Proc. of AH 2000. 131～143
10 Mobasher B,Cooley R,Srivastava J. Automatic Personalization based on Web usage mining. ACM Computer,2000,43(8):142～151
11 Yu P. Data Mining and Personalization Technologies. In:Proc. of DASFAA. 1999. 6～13
12 王继成，等. Web 文本挖掘技术研究. 计算机研究与发展，2000,37(5):513～520
13 王实，等. Web 数据挖掘. 计算机科学，2000,27(4),28～31
14 陈宁，等. 数据采掘在 Internet 中的应用 计算机科学，2000,26(7):44～53
15 Pei J,Han J. Mining Access Pattern Efficiently from Web Logs,Conf. of PAKDD'2000
16 http://www.w3.org/

(上接第41页)

```
2) receiveShadow:
 [A message ("MoblieShadow",shadow) has arrived]
 receive(("MoblieShadow",shadow));
  if (shadow.shadowTTL != 0)
    if (shadow.homePlace != place.name())
    {
      shadow.currentPlace = place.name();
      shadow.List.add(shadow);
    }
    else
    {
      //shadow comes back home
      surrogateID = shadow.shadowed;
      suurogate = shadowList.find(surrgateID);
      shadowList.remove(surrogate);
      shadowList.add(shadow);
      shadow currentPlace = Null;
  }
 ;
3) shadowProxyPathTimeOut:
 [The timer triggered a (timer,shadow) message]
 receive((rimer,shadow));
 shadowList.remove(shadow);
4) terminateShadow()
 if (currentPlace != Null)
  //Shadow moved
    send(currenrPlace,("Terminate",shadowID));
  delete(rhis);
5) receiveTerminate.
 [A message ("Terminate",shadowID) has been received]
 shadow = shadowList.find(shadowed);
 if (shadow != Null)
 shadow terminareShadow();
```

评价 基本影子协议、层次影子协议、移动影子协议完整地构成了移动 Agent 的影子控制协议。影子协议通过检测影子是否存在，来探测孤儿。

通过路径信息的帮助找到 Agent 的 Proxy。按照 Proxy 指示的信息，给目标场所发送查找消息，然后目标场所检查 Agent 是否在当地，如果不在，继续到下一个场所去找，直到找到 Agent 的场所，即找到路径的终点，实现了 Agent 的定位。

Agent 的 TTL 计时器决定何时终止 Agent 的执行。

总之，影子协议可以方便实现 Agent 的定位、Agent 的终止和孤儿的探测。

参 考 文 献

1 Genesereth M, Ketchpel S. Software Agenr. USA,1994
2 Shaham Y. Agent-oriented Programming. USA,1993
3 Baumann V. Conrrol Algorithms for Mobile Agent. University of Sruttgart,1999
4 Jennings N, Wooldridge M. Agent-Oriented Sofrware Engineering. University of London, UK, 1999
5 EURESCOM. Project712: Intelligent and Mobile Agents and Their Applicability ro Service and Network Management. Germany. 1999