基于 Agent 的平行迭代再工程方法研究与应用

陈志泊 程舒晗

(北京林业大学信息学院 北京 100083)

摘 要 随着软件技术的发展,更多的软件出现,维护和再工程的需求也更加迫切。通用的软件再工程的基本概念和模型无法保证再工程的高效低风险。创造性地将平行迭代模型与 Agent 技术结合起来,可以解决复杂遗留系统再工程问题。此方法在森林植被仿真系统 FVS 的再工程过程中,提高了项目的效率和成功率,并保持了目标系统的灵活性和可扩展性,取得了理想的效果,也为类似系统的复用和重构提供了参考。

关键词 再工程,Agent,平行迭代

中图法分类号 TP311.5 **文献标识码** A

Study and Application of Agent Based Parallel Iteration Reengineering Method

CHEN Zhi-bo CHENG Shu-han

(Department of Information, Beijing Forestry University, Beijing 100083, China)

Abstract Software needs to be maintained and reengineered, as the program techniques. With the study of the concept and models of software reengineering, an method was proposed to reengineer a massive legacy system combining parallel iteration and software agent technology. This approach was later applied in the reengineering of the FVS system, a FORTRAN program which is more flexible and scalable, and is a good example approach for other reengineering process of similar legacy systems.

Keywords Reengineering, Agent, Parallel iteration

1 引言

随着软件平台和开发技术的发展,以及系统需求的不断调整,软件在交付之后难免会面临结构退化、维护成本提高等问题。这些具有一定价值,但很难进行维护和进一步演化的软件称为遗留系统(Legacy System)[1]。遗留系统丰富的功能业务特征和领域特征决定了其重要的价值,它们通常不会被直接舍弃,因此开发和维护人员常常会面临需要对其进行软件变更的问题。软件变更是普遍存在的,Lehman 和 Belady 对于软件变更的必要性曾提出两条著名的定律:第一,任何投入使用的软件都必须随着需求不断变化,否则它的使用价值将会逐渐降低;第二,即使软件能及时变化,其结构也会逐步退化[3]。

为应对软件的需求变更和结构退化,在软件变更中不但要关注如何对遗留系统的功能模块进行修改,还需要研究对系统体系结构实施变更的方法,以避免系统在变更后太快再次失去价值,同时需要提高其可维护性。

对遗留系统的软件变更通常有两种方法:

1)重新设计开发新的系统。这种方法代价大、周期长、风险高,因此许多企业和组织会尽量避免采用这种方案。

2)对遗留系统进行再工程(Reengineer,也称作"再造"),即使用逆向工程、重构和正向工程技术将遗留系统整合到新的软件或硬件平台,具体过程如图 1 所示[4]。为方便叙述,本文将要实施再工程的系统定义为遗留系统,改造完成的系统称为目标系统;二者转换过程中的中间状态称为过渡系统。

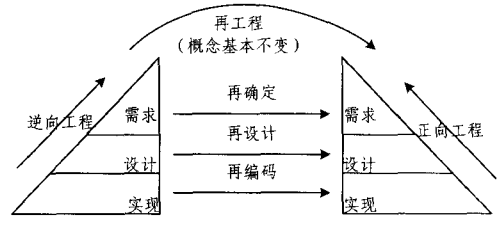


图 1 再工程的定义

本文主要探讨一种实用高效的大型复杂遗留系统再工程方法:首先运用逆向工程、软件分析技术对软件概念、业务逻辑和具体实现进行提取,重新划分模块,然后在正向工程中设计新的软件体系结构;并引入再工程过程模型以保证任务实施的进度更加平坦和高效,最终实现对遗留系统的重构。

2 基于 Agent 的平行迭代

大型复杂系统的再工程有 3 个阶段的问题:

1)逆向工程过程中,如何快速、准确地对庞大复杂的遗留系统进行分析和理解。

2)重构过程中,如何设计更好的目标系统,使其满足新需求的同时兼顾以后的可扩展性。这个过程中涉及到对目标系

到稿日期:2011-07-05 返修日期:2011-10-02 本文受国家林业局 948 项目(2008-4-48)资助。

陈志泊(1967—),男,博士,教授,博士生导师,主要研究领域为数据库技术、计算机软件与理论、嵌入式系统开发等,E-mail:zhibo@bjfu.edu.cn;程舒晗(1986—),男,硕士生,主要研究领域为计算机软件与理论。

统状态的理解、软件架构的设计，以及对正向工程技术的选择问题。

3)正向工程过程中，如何尽可能地缩短周期，降低风险，保证再工程顺利完成。

本文就以上问题，提出了基于 Agent 的平行迭代再工程的方法，其利用 Agent 思想进行程序理解和重新封装，提高逆向工程的效率以及目标系统的可维护性；使用平行迭代的过程模型对再工程的流程进行控制，提高再工程效率，降低风险，缩短周期。

2.1 基于 Agent 技术的再工程

软件开发技术的发展也推动了再工程技术的发展，在这方面人们做了大量的研究和实现。例如，借用软件复用技术中的构件概念，可以实现基于构件的再工程[6]；有将面向对象与构件概念相结合的研究[7]，及再工程结合面向服务[12]、面向性能[5]等方面的尝试。针对遗留系统开发语言陈旧的问题，还出现了代码翻译方法的研究[8]。

Agent 理论与技术最早出现于分布式人工智能领域，定义一个 Agent 至少必须具备反应性、自治性、面向目标性和针对环境性 4 个特性，在此基础上还可以包括许多其他的特性[9]。面向 Agent 的软件工程认为系统都是由一个或者多个 Agent 构成；Agent 之间通过合作、协商、竞争等交互关系协同实现系统的整体设计目标。面向 Agent 的再工程方法[15]能够将二者系统紧密地结合起来，平稳地从逆向工程过渡到正向工程。本文将结合再工程的定义以更细的粒度来进行分析说明。

以下将讨论两个过程中常见的技术和方法，分析 Agent 技术用于再工程的优势和不足，并给出基于 Agent 再工程的目标系统设计。

2.1.1 逆向工程方法

逆向工程技术对程序理解和分析主要有 3 种方法：1)自底向上：从机器码出发，用反汇编等技术分析程序，其主要适用于缺乏源代码和文档的遗留系统；2)自顶向下：从系统的问题域到具体实现域构造映射集，从程序语义层去理解，其适用于对开发领域非常熟悉的再工程；3)二者的结合模式：兼顾了两种方法的特点，从程序源码出发，使用程序分段、切片等方法进行程序理解，其适用于具备源码和少量文档的遗留系统。此外，根据不同标准，存在着各种软件分析的分类方法。例如，从方法学上可以将软件分析分为面向过程的、面向对象的、面向构件的等；从分析对象上，可以将其分为对代码、模型、文档等的分析；根据分析过程中是否需要系统运行，可以将其分为静态分析和动态分析。

常用软件分析和理解方法对软件理解能够起到一定的辅助作用，但是仍存在着可操作性不强、结果不直观等问题，尤其是对大型软件系统缺乏好的支持[13]。文献[13]提出了基于多 Agent 的程序理解方法模型，其分为数据收集与管理、可视化逆向建模和结果分析与细化 3 个阶段(图 2 对应的 1、2、3)对软件进行分析和理解。通过对系统进行 Agent 化，把 Agent 的通信、谈判、协调等功能引入遗留系统，充分利用 Agent 的自主性、智能性和自适应性来更好地理解系统。该方法采用静态分析和动态分析结合的方式，基于静态分析提取源码特征，然后根据动态分析从系统运行模式中提取 Agent，理解程序。本文对这个过程进行了改进，其结构如图 2 所示。

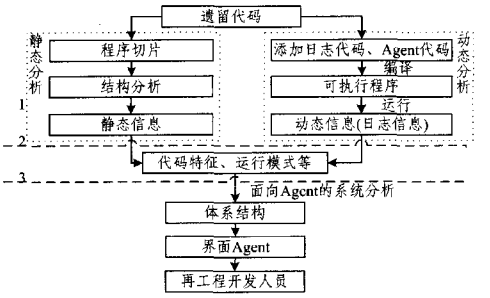


图 2 基于 MAS 的程序理解系统结构图

在静态分析阶段，可以采取常用的软件分析方法，主要是切片分析和结构分析。程序切片法的主要思想是，通过分析源码的语法关系来提取程序中功能相关的代码，形成新的模块(切片)；结构分析法则试图通过分析程序内部的调用关系图、类图等对程序的结构进行分析，以帮助开发人员理解系统。

在动态分析阶段，需要在遗留代码中添加一些日志代码和 Agent 代码来跟踪获取系统的动态信息，从“事件踪迹”中更好地理解软件的功能和运行模式。Agent 代码是指与 Agent 相关的辅助代码，用来辅助对 Agent 模块划分的确定过程。这个过程可以认为是从实现和功能方面去理解系统的业务逻辑，对结构进行分析和抽象。

值得说明的是，对于包含面向过程、面向对象代码的遗留系统，提取组件信息通常可以分步进行：1)先从面向过程的遗留代码中提取对象信息。通过研究函数、函数参数、全局变量之间的关系，可以将函数和变量聚类到统一的抽象分组，从而获得对象信息。2)从面向对象的遗留代码或者对象信息中提取组件信息，即使用软件聚类的方法根据对象信息划分组件[12]。此外，在进行程序分析的过程中，还需要留意一些影响性能的“反模式”设计，并在重构过程中将其消除掉[5]。

2.1.2 重构和正向工程方法

逆向工程完成了对软件的理解，正向工程则需要对其进行分析和重构。

对不是很古老的遗留系统实施软件变更时，往往只需要涉及功能模块的重构，但模块级的软件变更无法避免系统结构的退化。系统最初的软件架构设计往往发生在细节需求尚未完成时随着开发和维护的展开，对软件需求的理解相对细化或者需求本身发生了变更，原先的架构可能会出现不足甚至不适应的地方。这种情况下需要根据新的需求，对软件的体系结构进行调整，甚至使用更先进的架构进行重新设计。因此，大型遗留系统的再工程往往涉及到软件体系结构的变更。

软件体系结构概念的提出最初是为了将软件需求与软件设计衔接起来，以解决软件系统的结构和需求向平坦过渡的问题。而软件体系结构技术的发展，促进了软件复用从代码复用发展到设计复用和过程复用。再工程中使用的软件体系结构需要满足两个目的：方便下一次功能的调整，防止系统结构过快退化。

常见的方案是使用面向服务的体系结构(SOA)和 Web Service 技术解决遗留系统改造、软件复用问题[12]，这种再工程的方式非常适合将遗留系统再工程到分布式的、基于网络的平台上，而且能有效利用遗留代码，提高开发效率、降低风险。但并不是所有的遗留系统都有分布和网络的需求，例如

单机遗留系统的再工程中，这种以牺牲系统性能[14]为代价带来的好处是不划算的。

文献[15]提出了采用多 Agent 系统(最好给出 MAS 的完整英文，MAS)技术对遗留系统进行包装的方法。首先将遗留代码封装为能与外部组件通信的组件，并引入推理层和通信层，将组件组装成为功能独立的智能 Agent；然后将得到的 Agent 通过 Agent 组件、Agent 连接件、Agent 约束等概念组合起来，用松散的结构、较低的成本，协同完成集中控制所不能完成的问题求解。系统如图 3 所示[15]。

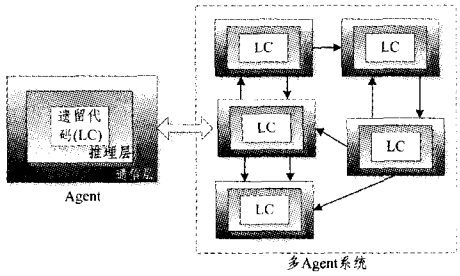


图 3 “包装器”包装的多 Agent 系统

在这样的 MAS 架构下，可以方便地将一部分具有共同特征的模块封装成为独立的 Agent。例如，可以把需要频繁与其他组件通信的部分实现为移动 Agent[16]，通过动态灵活的系统组成运行时(runtime)，将移动 Agent 向需要交互的组件移动来避免对网络带宽的依赖。Agent 技术重构使得目标系统能够适应网络或者分布等新环境，能迅速给系统加入新需求，具有很好的扩展性。

2.1.1 节中讨论的基于 Agent 逆向工程方法，给正向工程的过程提供了提取出来的组件模块，需要结合新的设计和结构对这些软件理解得到的 Agent 进行分析和完善。有了逆向工程时的工作基础，对目标系统进行基于 Agent 的正向工程也就非常顺利和自然了。

2.1.3 基于 Agent 技术的目标系统

使用 Agent 技术进行再工程，最终生成的目标系统状态是再工程中非常重要的内容。文献[16]给出了一种应用 Mobile Agent 技术再工程得到目标系统的方法。本文主要的研究对象是单机遗留系统，因此简化网络环境的迁移，扩展封装对象的范围，设计得到的一个目标系统，如图 4 所示。

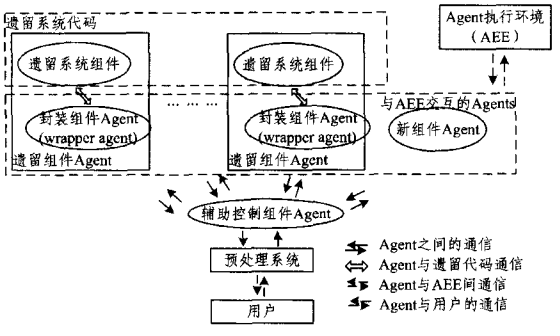


图 4 基于 Agent 再工程的目标系统体系结构

用户不需要了解再工程的内容，因此引入了预处理系统来解决用户接口的问题。用户请求和输入数据经过预处理系统，将传递给辅助控制组件 Agent 进行调度和处理。具体的业务逻辑则由遗留组件 Agent 和新组件 Agent 来完成。遗留组件 Agent 由封装组件 Agent 和遗留系统组件构成。封装组件 Agent 能够最大化地重用遗留代码，并使其具有 Agent 的特性。该目标系统体系结构还涉及到多 Agent 系统(MAS)通信的问题，这方面的研究比较成熟，篇幅所限，这里不做叙述。

2.2 再工程过程模型及框架

前面讨论了再工程过程中逆向工程、正向工程的方法，本节重点关注遗留系统再工程的过程模型，用更细致的方法来指导项目的具体实施。

2.2.1 再工程替换策略

再工程的过程模型主要研究的是系统的新旧替换策略，可以分为：1)一次性替换法，即全面分析遗留系统之后，设计全新系统来替换整个系统；2)迭代式替换法，主张从系统结构的角度出发逐步替换遗留系统中的组件；3)演化式替换法，主张从系统功能的角度逐步替换组件[11]。后两种替换法也可以归纳为局部替换法，主要思想是通过“少量多次”的方法对遗留系统的模块进行替换，二者的主要分歧在于迭代替换是基于语法上的模块划分，而演化替换是基于逻辑上的功能划分。很明显，对于大规模的复杂遗留系统，一次性替换法需要一次性对系统进行完全重构，风险很高而且周期很长，因此很少被实际采用。演化式替换法也不够实用，因为软件系统中的许多逻辑功能是由多个组件协同完成的，业务逻辑提取、解耦合和分批替换的过程将会非常复杂。因此迭代替换法是最常用的替换策略。

再工程的方法大致可以分为 3 种，如图 5 所示。大爆炸(Big-Bang)方法是指重新设计新的目标系统，增量式替换是根据需求逐步增加或修改已有模块，演化式替换则是重新划分模块，对系统逐步替换。这 3 种方法并不是严格区分的，它们可以结合实施。平行迭代模型就是基于后两者替换方法发展起来的。

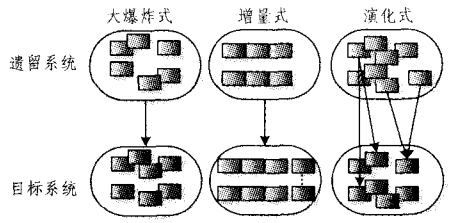


图 5 再工程的 3 种方法

2.2.2 平行迭代模型

Alessandro 等于 2003 年系统性地提出了迭代再工程模型和方法[10]，其主要优势有：1)通过“分治”的思想，将再工程需要处理的系统维度降低，以易于管理和实现；2)对遗留组件的改造过程中可以增加新组件；3)系统中的新旧组件可以并存，减少了新旧系统替换所需的“冻结”时间，系统在再工程过程中可以不间断地提供服务。为了在迭代再工程中加快进度，减少多次迭代的开销，文献[11]对 Alessandro 的模型进行了分析和改进，提出了平行迭代再工程模型，并给出了形式化的证明。其主要的改进在于将遗留组件的改造从单次单个组件变成了多个遗留组件平行改造。本文将 Agent 思想应用到平行迭代模型中，其流程如图 6 所示。

如图 6 所示，迭代再工程模型首先要求对系统进行分析，包括对分类数据、重新设计数据库、平行迭代分析等，以便对遗留系统进行数据库移植和组件重组。使用基于 Agent 技术的逆向工程对软件进行分析，能加快逆向工程的速度，并且为

后期的重构提供初步的 Agent 模型。迭代过程中新、旧数据库的同步访问是模型的重点，因此提出了将数据访问重定向的方法来解决这个问题，元数据就是重定向的中间产物。等价性测试是迭代模型中用来保证遗留组件的改造符合要求的重要手段，这个过程将不断重复直到系统改造完成。除了允许多个组件同时改造，平行迭代模型对传统迭代模型的另一个改进在于引入了“非等价因素定位”的辅助过程。这样就允许在迭代中对定位到的不等价因素进行改造，使模型更全面。平行迭代模型对迭代模型的改进，对于工期紧、数据存储访问要求高的再工程来说尤其有意义。

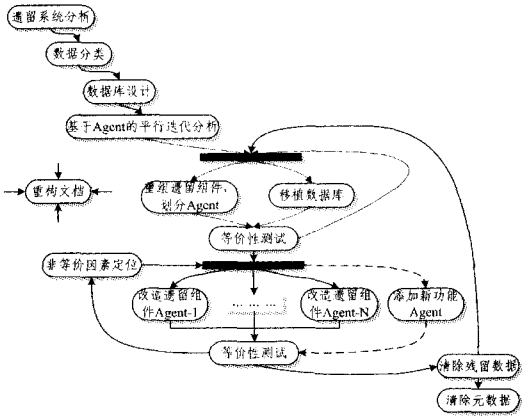


图 6　基于 Agent 平行迭代再工程流程图模型

将平行迭代模型应用于遗留系统的再工程的好处是非常明显的。首先，迭代再工程模型降低了再工程的风险，通过多次迭代去改造，降低了迭代的难度。其次，提高了每次迭代对系统修改的可控性，分析改造引入的开销、错误等都更加容易，整个过程的管理更加方便。再次，平行迭代使得一些逻辑相似的模块不必等到下一个周期再改造，这样能够缩短再工程的工期。最后，平行迭代模型更加方便目标系统引入新的功能。

将 Agent 技术应用于平行迭代，重点在于遗留组件向 Agent 的转化。遗留组件是迭代再造模型的基础，因此在进行软件分析时需要从遗留系统中提取出组件信息。结合 2.1 节中的讨论，可以将提取出来的组件信息封装和转换成为不同的 Agent 模块。遗留组件改造成为 Agent，主要是通过 2.1.3 节中提到的封装 Agent(wrapping agent)辅助实现的。

通过以上分析可知，平行迭代模型与基于 Agent 技术的再工程建模保持一致。通过基于 Agent 的逆向工程、正向工程技术保证目标组件 Agent 的正确性，平行迭代过程模型可控制整个再工程过程的进度和内容，能有效利用遗留代码，降低重构风险，提高工程效率。

3　实例分析：FVS 再工程

案例采用的是森林植被仿真系统(Forest Vegetation Simulator，FVS)的再工程。

3.1　FVS 再工程的要点

FVS 是一个具有近 40 年历史的大型复杂遗留系统，代码近 60 万行，包括 FORTRAN、C、SQL 等多种实现语言。FVS 的主要功能是预测和仿真美国森林植被的生长和产出情况。

FVS 中的林业模型全部以硬编码的形式固化在代码中，而这些林业模型与地理区域、树种等信息密切相关。这种实现方式使其无法适用于中国的地理条件和树种类型，除非对林业模型涉及的模块进行重构。FVS 所涉及到的林业模型很多，在代码级将中国的模型应用到程序中的工作将是低效而且繁冗的，后续的维护也同样复杂。因此，必须将计算公式从系统中尽可能地独立出来，使得程序可以在运行过程中从外部的模型数据库中读取并解析公式，再将公式应用于计算；林学专家甚至提出了运算中自行校准的公式，以及将其保存到模型库中的需求。

根据上述需求，外部可编辑的模型数据库也是目标系统的必要组成部分。此外，目标系统还需要具备较好的可扩展性，以便使用类似插件的形式对系统功能进行完善和扩充。

篇幅有限，对于 FVS 的再工程不做展开描述。

3.1.1　FVS 的逆向工程

摘取出来的代码，结构是比较松散的，需要我们根据程序分片得到的模块关系，组织工程结构，建立子项目。结合部分 FVS 说明文档和 Understand for FORTRAN 程序的组织，可以将 FVS 的调用做程序切片。抽象简化得到的最终系统体系结构模型如图 7 所示。

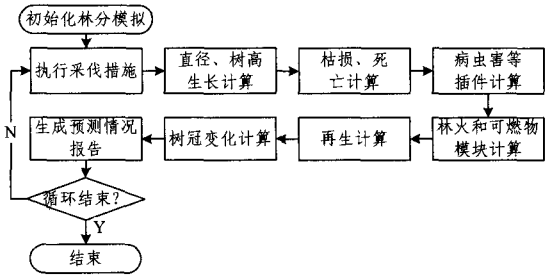


图 7　FVS 简化遗留体系结构

3.1.2　FVS 的正向工程

通过逆向工程对系统的理解，结合林学专家提供的领域特征，最终要实现的系统模型如图 8 所示。

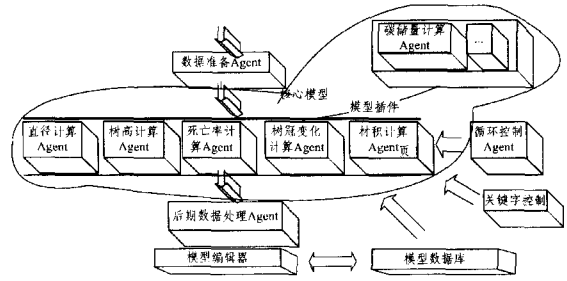


图 8　FVS 再工程目标模型

将程序分片对系统划分得到的模块，使用 Agent 技术重新封装和组织。考虑到林学模型的复杂性，为了降低风险和可控性，对于预测和仿真过程中不同阶段的计算分别采取封装不同 Agent 的方案。林业模型外围的数据准备等工作利用遗留代码完成，计算公式的动态解析则用 FORTRAN 调用新加的公式解析代码进行处理。Agent 采用 Java 语言封装，由于 Java 与 FORTRAN 不能直接混合编程，因此还需要使用 C 语言对 FORTRAN 进行封装。文献[14]对需要注意的问题有比较清楚的说明。

3.1.3　FVS 平行迭代

如图 8 所示，我们对直径计算 Agent、树高计算 Agent、死亡率计算 Agent、树冠变化计算 Agent 和材积计算 Agent 进

行平行迭代改造，其他类型的 Agent 在另外的阶段进行迭代。其中跟计算相关的 Agent 也分组放到平行迭代的周期中，以在加快工程进度的同时保证较低的风险。封装的智能 Agent 可以很方便地与新添加的模型数据库交互，对新增加的计算 Agent 仅通过修改循环控制 Agent 就能完成，实现了良好的可扩展性。

3.2 FVS 再工程过程评估

对 FVS 的再工程工作始于 2006 年，项目组最初的方法和思路是通过逆向工程的方法分析和理解系统，重新设计架构，在 Windows 平台下采用 Javascript 结合 DOM 技术的方法来实现目标系统。但是经过开发人员的几次变更，再工程进度受到严重影响，最终目标系统变成了新的遗留系统。本文提出的模型则可以有效地避免这样的问题。

本项目组前期对 FVS 实施了传统的再工程，得到了具备 FVS 基本工程的目标系统，而且再工程期间编写的文档使得目标系统具有较好的可维护性。但是由于相关人员毕业，再工程的主要实施人员流失，新加入的开发人员对目标系统的理解需要花费时间，而且涉及对遗留系统的剩余功能的持续再工程，所以对系统进一步扩展的难度较大。另一方面，FVS 是从美国引进的系统，美国林务局的开发人员仍然在对该系统进行更新和维护，这就对我们的目标系统可扩展性提出了更高的要求。传统再工程方法得到的目标系统是无法比较直接地应用遗留系统的这些变化和更新的，因此扩展性不佳的目标系统肯定会存在相当程度的滞后。

基于 Agent 进行逆向工程时，对于许多遗留代码的细节可以不必深入理解，研究的重点更多的是方法和模块的接口与用法，从而为开发人员节省了大量的时间。有了逆向工程和重构过程中整理得到的模型结构，大部分的代码可以直接或间接地加以利用，与传统再工程方法相比，本方法中正向工程的时间大大缩短了。由于尽可能地利用了原有代码和接口，因此生成的目标系统也能比较方便地应用由美国林务局维护和变更的 FVS，且可扩展性很好。

表 1　基于 Agent 的平行迭代与常规方法在 FVS 再工程中的对比

再工程方法	参与人数	逆向工程时间	设计重构时间	正向工程时间	再工程完成度	目标系统可扩展性	目标系统可维护性
常规再工程方法	8～10 人（包括 3 名林业专家）	6 个月	1 个月	6 个月	差	较差	较好
基于 Agent 平行迭代再工程	3～5 人（1 名林业专家）	4 个月	1 个月	2 个月	较好	好	较好

表 1 是将基于 Agent 的平行迭代方法与常规方法在 FVS 再工程中的应用进行的比较。第二次再工程中逆向工程的过程虽然是新的开发人员，但是使用了前期的一些成果，因此这部分的时间不做比对。对于实现同样功能的目标系统，基于 Agent 的平行迭代方法正向工程的工期仅为传统再工程方法的 33%，而且实现了更好的可扩展性。可以看出，本文提出的方法所带来的好处是十分明显的。

目前 FVS 的再工程完成了对直径计算 Agent、材积计算 Agent 以及碳储量计算 Agent 的迭代再工程，实现了第一次再工程中目标系统的全部功能。虽然对其他 Agent 的迭代再工程尚未完成，但是这些工作能从目前的研究中借鉴大量的经验，因此其可行性和难易程度是乐观的。

结束语　软件学科主要研究 3 个层次的问题[2]：一是软件的本质和模型；二是特定软件模型下的开发技术和方法；三是特定软件的具体开发、应用和维护。对应地，关于软件变更问题的研究可以分两个方向：1）正向研究，即从软件本质和模型角度出发，在系统设计阶段就充分考虑到软件变更的可能性，例如基于构件、基于复用的设计等方法；2）逆向研究，即从软件维护和再工程角度出发，主要研究现存系统的软件变更方法。

尽可能利用已有资源，更快地开发出更高质量的软件，是软件工程研究的主要目标。从开发的角度，基于复用的软件开发、基于构件库的开发、基于构件的体系结构等研究课题都是围绕着这个目标所开展的，开发者从遗留系统再工程的角度也做了大量的研究和实践工作。常见遗留系统再工程方法主要有：将遗留系统以 SOA 的形式进行模块划分和服务封装[11,12]等、将 Agent 技术和思想应用于遗留系统[13,15,16]等。单独应用 Agent 技术和方法改造遗留系统，缺乏合理的过程模型做指导，再工程的风险大、周期长；仅使用平行迭代方法再工程，虽然解决了风险和周期的问题，但是缺乏程序理解和体系结构设计的理论基础和关键技术。

本文探讨了遗留系统再工程中最重要的 3 个问题，即再工程的过程模型、逆向分析中的软件分析方法、正向工程中软件体系结构的重新设计方法，创造性地将平行迭代再工程模型与 Agent 技术相结合，提出了基于 Agent 的平行迭代再工程模型，并以森林植被仿真系统 FVS 为例对模型进行了验证。本文的实践为类似系统再工程的研究和实践提供了参考，同时为进一步实现灵活可扩展的森林植被系统奠定了基础。通过对 FVS 系统再工程研究过程的描述可以看出，Agent 技术保证了目标系统的灵活性和可扩展性，平行迭代模型降低了 FVS 再工程的风险和难度，达到了良好的效果，也提供了将遗留系统改造成 SOA 系统等常用解决方案之外的另一个思路。

参 考 文 献

[1] 刘晓建，陈平，蔡希尧. 遗产系统及其解决方案的综述[J]. 计算机科学，2002(5)：127-130

[2] 杨芙清，梅宏，吕建，等. 浅论软件技术发展[J]. 电子学报，2002(S1)：1901-1906

[3] Lehillan M M, Belady L A. Program Evolution: Proeesses of Software Change[M]. Aeademie Press, 1985

[4] 郭耀，袁望洪，陈向葵，等. 再工程——概念及框架[J]. 计算机科学，1999(5)：78-83

[5] 沈钺，彭鑫，夏宽理，等. 面向性能的软件再工程研究[J]. 计算机工程，2005(3)：7-9

[6] 药锐，赵文耘，张志. 遗产系统的构件化技术[J]. 计算机工程，2004(8)：48-50

[7] 刘冬懿，李虎，金茂忠，等. 遗留系统再工程中交叉构件划分方法[J]. 北京航空航天大学学报，2005(10)：80-84

[8] 杨卫平，赵合计. 遗产软件的代码翻译[J]. 计算机工程，2004(6)：83-85

[9] 刘大有，杨鲲，陈建中. Agent 研究现状与发展趋势[J]. 软件学报，2000(03)

[10] Bianeh A, Caivano D, Mareng V, et al. Iterative Reengineering of Legaey System[J]. IEEE Transactions on Software Engineering, 2003, 3(29)：225-241

（下转第 160 页）

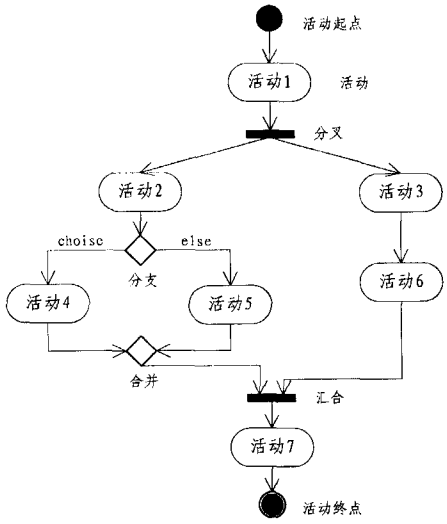


图 6　UML 活动图示例

依据本文所提出的原语-模式的映射规则，构建 UML 活动图建模原语-模式与 OV-5 元模型概念元素之间的语义映射关系，如表 3 所列。

表 3　基于原语-模式 UML 活动图与 OV-5 元模型映射规则

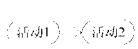
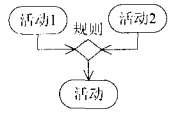
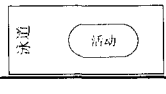
UML 活动图建模原语-模式	语义	对应 OV-5 元模型的概念元素
模式(Patterns)		
	顺序模式：表示活动的顺序执行	Information/Resource；Activity；activityConsumesResource；activityProducesResource
	分叉模式：用来描述活动的并发执行，每个分叉有一个输入，两个或多个输出转换	Information/Resource，Activity；activityConsumesResource；activityProducesResource；activityBeforeAfter
	汇合模式：用来描述多个并发控制流同步发生，当所有活动都达到汇合点后，才能继续执行以下的活动	Activity，activityConsumesResource；activityProducesResource；activityBeforeAfter
	分支模式：描述一个有条件的活动执行过程，即用一个布尔表达式的真假来判定需要执行的活动	Condition；Rule；Activity；RuleConstrainsActivity；ActivityPerformableUnderCondition；activityBeforeAfter
	合并模式：表示从对应的分支开始的条件行为的结束	Activity；RuleConstrainsActivity；Rule；ActivityBeforeAfter
	泳道-活动模式：描述活动由执行者来执行	Activity；Performer；activityPerformaedByPerformer

表 3 中裁剪了 UML 活动图的建模模式，6 种模式可以满足 OV-5 的建模需求(见 3.1 节)，并扩展了 OV-5 描述的内容，为实现体系结构模型的多重使用提供了基础，从而规范了基于 UML 活动图的 OV-5 建模，提供了精确一致的体系结构描述。限于篇幅，本文不再对基于 UML 活动图建模的 OV-5 模型数据 XML 导入、导出接口进行设计，具体实现可参照算法 1。

结束语　当前国内外主流的体系结构建模工具有 System Architecture(SA)、Enterprise Architecture(EA)、Rational Rose 以及国内相关研究单位(包括国防科技大学、中电 28 所、解放军理工大学等)开发的体系结构建模工具，它们所支持的框架方法论及采用的建模方法和原理都不尽相同。为更好地支持体系结构的理解、比较和集成，关于实现不同框架、不同工具及建模方法下所开发体系结构之间共享和交互的需求显得越来越迫切。本文基于 XML 的模型转换方法，分析了方法中涉及到的 3 种转换规则，重点研究了建模语言原语-模式与元模型数据元素严格的语义映射关系，有效地维护了底层数据的规范性、一致性，降解了不同体系结构工具、建模方法由于选用框架及设计原理上的不一致而引起的语义冲突。并且，本方法不强制用特定的建模方法进行体系结构建模，如 OV-5 建模可以选择 IDEF0 建模方法或 UML 活动图建模方法。基于本方法进行体系结构建模，能有效支持体系结构数据的重用，提高体系结构开发的效率，为支持多种框架、多种建模方法的体系结构建模方法提供了良好的理论基础。

参 考 文 献

[1] DoD Architecture Framework Working Group，DoD Architecture Framework Version 2.0[R]. U S:Department of Defense，2009

[2] UK Ministry of Defense. UK Ministry of Defense Architectural Framework(MODAF) v1.2.004[R]. UK Ministry of Defense，2010

[3] Bailey I，Partridge C. Working with Extensional Ontology for Defence Applications[C]//Ontology in Intelligence Conference. 2009

[4] 罗雪山，罗爱民，张耀鸿. 军事信息系统体系结构技术[M]. 北京：国防工业出版社，2010

[5] 罗爱民. 基于框架的 C4ISR 体系结构语法、语义设计与分析方法研究[D]. 长沙：国防科学技术大学，2006

[6] OMG. OMG XML metadata interchange specification version 2.1 [EB/OL]. http://www.omg.org/cgibin/doc? formal/2005-09-01.pdf，2005

[7] 李清，李伟明，徐大丰. 基于元模型的企业模型表达[J]. 清华大学学报，2008，48(7)：1209-1212

[8] 王杏林，曹晓东. 概念建模[M]. 北京：国防工业出版社，2007

[9] 严悍，刘冬梅，等. UML 软件建模：概念、规范与方法[M]. 北京：国防工业出版社，2009

(上接第 132 页)

[11] 李珏峰. 基于量化方法的大型遗留系统迭代再造研究[D]. 杭州：浙江大学，2008

[12] 许鹏. SOA 架构下 Web Services 实现的企业遗留系统重用研究[D]. 合肥：合肥工业大学，2008

[13] 王少锋. 基于多 agent 的程序理解方法研究[J]. 计算机科学，2002(5)：131-133

[14] 汪小林，邓浩，王海波，等. Fortran 地理模型的拆分与服务化封装[J]. 计算机科学与探索，2011(3)：221-228

[15] 钱佳. 基于 MAS 技术遗留系统包装器体系结构风格的研究[D]. 太原：太原理工大学，2006

[16] 詹剑锋，程虎. 基于 Mobile Agent 技术的遗留系统再工程方法[J]. 软件学报，2002(12)：2343-2348