

业务流程驱动的森林仿真构件组装技术及应用研究

董天阳 李文杰 范 菁 夏佳佳 熊丽荣

(浙江工业大学计算机学院 杭州 310014)

摘 要 针对森林仿真系统的需求或应用目标不断发生变化的问题,提出了一种业务流程驱动的森林仿真构件组装方法。该方法设计和实现了森林仿真领域的构件模型和仿真构件接口,并通过业务流程驱动的构件组装框架实现了森林仿真系统的业务流程、模型算法等的软件复用,解决了现有森林仿真系统存在的快速构建或重构困难的问题。通过在虚拟森林仿真系统开发中的应用表明,基于构件组装的方法与代码级重用、重新开发方式相比,可以大幅减少开发工作量,降低系统的开发难度,实现森林仿真系统的快速搭建或重构。

关键词 森林仿真,软件构件,构件组装,流程驱动

中图分类号 TP391.41 **文献标识码** A

Process-driven Method of Component Assembly for Forest Simulation System and its Application

DONG Tian-yang LI Wen-jie FAN Jing XIA Jia-jia XIONG Li-rong

(College of Computer, Zhejiang University of Technology, Hangzhou 310014, China)

Abstract Due to the changing of the requirement and the application aim of the forest simulation systems, this paper presents a process-driven method of component assembly for forest simulation system. This method designs and implements the component models and the interfaces of forest simulation components, and reuses the business processes and computation algorithms by assembling software components. It can solve the existing problem that the forest simulation system is difficult to quickly build or reconstruct. In addition, this method is successfully applied to develop the forest simulation systems, and it can achieve rapid construction and reconfiguration of forest simulation systems. The applications show that the process-driven method of component assembly for forest simulation system proposed in this paper can significantly reduce development cycle and development difficulty compared with other methods, such as code-level reusing, re-development.

Keywords Forest simulation, Software component, Component assembly, Process-driven

1 引言

森林生长的动态变化具有复杂度高、时空跨度大等显著特征^[1],利用计算机技术模拟森林场景是一项颇具挑战性的工作。目前国内外对森林仿真的研究大多集中在森林场景中实体的建模可视化技术以及各种模型和算法的优化上,比较注重仿真的效果,但对仿真系统架构的关注相对比较少,尤其是对支持多粒度的动态森林场景体系结构设计问题的研究更少^[2]。传统的仿真系统往往是集中式的,结构固定、缺乏交互,而且不易于扩展。近年来,国内外研究机构在仿真系统架构方面也开展了新的探索和初步实现^[3-5],但仍处于起步阶段。

基于构件的软件开发是 20 世纪 90 年代提出的,是一种理想的软件开发理念,目的是实现软件复用和构件化生产,从而节约系统的开发时间和开发成本。目前,基于构件的软件

开发方法在农林业相关领域受到越来越多的关注。总的来说,现有的构件技术主要用于“森林资源信息管理”、“林业决策系统”、“农业专家系统”中^[6,7],或仅仅是用于构建植物三维模型^[8-11]。由于森林仿真涉及较复杂的形态和生态的处理,目前将构件技术完全引入森林仿真系统架构的研究还很少。

此外,不同机构进行森林仿真的应用目标和出发点往往不同,在时空尺度上可以分为单木、林分和全林分 3 种不同粒度。目前,多数森林仿真系统都是针对各自特定的研究目标,采用对应的生长模型或仿真技术来进行单一粒度的森林生长仿真,系统往往都是针对个性化需求单独设计和开发的。不同粒度的森林场景都需要构建地形和天空等场景实体,而且单木仿真生成的树木模型可以被重复用于林分和全林分仿真中,还有诸如纹理映射等算法不断重复出现在不同系统中,但是这些系统之间缺乏关联和映射,因此它们很难直接复用彼

到稿日期:2011-11-04 返修日期:2012-03-09 本文受国家自然科学基金项目(61173097),浙江省重点科技创新团队项目(2009R50009),浙江省重大科技专项(2009C11027),浙江省科技计划项目(2010C33046),浙江省自然科学基金项目(Z1090459, Y1101102)资助。

董天阳(1977-),男,博士,副教授,CCF 会员,主要研究方向为虚拟现实和软件构件技术;李文杰(1985-),男,硕士生,主要研究方向为虚拟现实和中间件技术;范菁(1969-),女,博士,教授,CCF 理事,主要研究方向为虚拟现实及可视化、软件中间件技术, E-mail: fanjing@zjut.edu.cn (通信作者);夏佳佳(1987-),女,硕士生,主要研究方向为虚拟现实及其可视化技术;熊丽荣(1973-),女,硕士,副教授,主要研究方向为软件构件技术。

此的模块。另外,由于多数森林仿真系统架构是基于传统的集中式架构,系统结构相对固定,森林仿真的业务流程和模型算法都被采用某种编程语言硬编码在各个系统中,因此该系统模块之间耦合度高,重构能力比较弱,无法快速地响应仿真需求的变更。

因此,本文基于构件思想架构森林仿真系统,设计了森林仿真构件模型和构件接口,提出了一种业务流程驱动的森林仿真构件组装方法,并将其应用于虚拟森林仿真系统的构建。该方法将仿真系统中业务模块构件化,实现了基于构件的森林仿真快速搭建或重构,提高了仿真软件的复用性,降低了森林仿真系统的开发难度。

2 森林仿真构件的设计

森林仿真构件的设计是构件化森林仿真系统设计中的关键技术,因为只有通过仿真构件的组合形成业务构件才能完成森林仿真。仿真构件的设计借鉴面向对象技术的思想,将具体的森林仿真业务逻辑模块进行封装,隐蔽了其内部技术实现的细节,封装后的模块提供了一组公开的输入和输出接口。构件作为仿真系统中可替换的模块,降低了森林仿真系统的耦合度和生态模型的复杂程度。

构件的设计应遵循“抽象、逐步求精、信息隐蔽、功能独立”等思想和原则,构件的可重用性、互操作性、可扩充性、易移植性、通用性和可变性等构件特性成为必须充分考虑的问题。如图1所示,一个构件的设计主要包括构件规格、构件接口和构件实现3个重要部分。

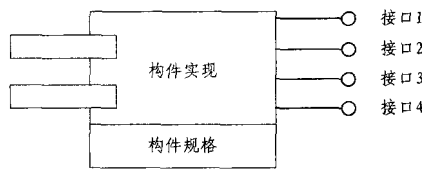


图1 构件示意图

2.1 仿真构件模型

构件规格(Component Specification),也称为构件规约,用于描述构件的特征,及如何使用和管理构件。构件模型作为构件精确描述的基础,主要是对构件本质特征以及构件间关系的抽象刻画,将对创建和实现构件起指导作用^[12-14],因此,建立合适的构件模型是实现构件化复用的第一步。本文参考3C模型、REBOOT模型、青鸟模型等经典模型以及计算机软件构件复用属性规范^[15],并结合森林仿真的具体特点,提出了针对森林仿真的构件模型。

定义1(仿真构件, SIMULATION COMPONENT) 森林仿真中能够独立完成一定仿真业务功能的可复用成份,通过构件基本信息、仿真语境、构件接口、构件实现、构件关系等方面来描述构件。

仿真构件::= \langle 构件基本信息,功能语境,构件接口,构件关系,构件实现 \rangle

SIMULATION COMPONENT ::= \langle COMPONENT_BASICINFO, SIMULATION_CONTEXT, COMPONENT_INTERFACE, COMPONENT_RELATION, COMPONENT_IMPLEMENTATION \rangle

定义2(构件基本信息, COMPONENT_BASICINFO) 主要是对构件最基本信息进行描述,包括构件名称、构件作者、构件版本、提交日期、缩略图、详细描述等。

COMPONENT_BASICINFO ::= \langle COMPONENT_NAME, COMPONENT_AUTHOR, COMPONENT_VERSION, SUBMISSION_DATE, COMPONENT_THUMB_NAIL, DETAIL_INTRODUCTION, MEMO \rangle

这些基本信息是构件的说明性信息,每个属性都可以用“属性/值”对来表示,仿真构件的名称应能比较直观地概括构件的特性;由于森林仿真的很多模块都涉及到图形图像元素,因此增加了缩略图来直观地预览该构件;详细描述了该构件所具有的功能、性能及其他可用性。

定义3(构件仿真语境, SIMULATION_CONTEXT) 主要包括使用该仿真构件必须满足的前置条件和后置条件,以及该构件在整个仿真流程中所完成的业务功能及部署配置。

SIMULATION_CONTEXT ::= \langle PRE_CONDITION, POST_CONDITION, POST_CON_STATUS, BUSINESS_FUNCTION, DEPLOY_CONFIGURATION \rangle

仿真语境的描述对整个仿真的进行起着关键作用,因为当载入某些构件时,如果其后置条件不满足,将影响整个仿真过程的进行。所以可以通过后置条件的状态进行判断,如果其后置条件不满足,就需要重新选择其他构件。例如,树木几何模型构件的后置条件需要绘制构件来进行可视化,如果当前绘制构件不存在或者不工作,将无法呈现仿真效果。本文将基于流水线式的构件加以组装,前置条件描述使该仿真构件正常工作已经满足的条件,后置条件的状态一般由其相关的构件关系来确定。

BUSINESS_FUNCTION ::= \langle MODELING \rangle | \langle CALCULATION \rangle | \langle RENDERING \rangle

业务功能将详细地刻画该构件目前在整个森林仿真流程中所处的位置及其完成的业务功能,比如完成树木三维网格的建模、树木模型的导入、纹理映射等功能。每个构件的功能类型将通过剖面分类的方法供用户在提交构件时进行选择性描述,这个重要的信息将用于构件组装过程中流程环节的识别。

定义4(构件接口, COMPONENT_INTERFACE) 描述应该如何与构件交互,是仿真构件对外行为的描述,这里隐藏了仿真构件内部的具体实现细节,主要包括接口名称、接口类型、属性集合和操作集合等。

COMPONENT_INTERFACE ::= \langle INTERFACE_NAME, INTERFACE_TYPE, ATTRIBUTION_SET, OPERATION_SET \rangle

实际上,接口可以独立于任何对其实现的构件而存在,即一个接口可以由不同人员用不同技术来实现,这种接口和实现的分离有助于利用构件技术来架构松耦合的森林仿真系统。接口类型通常包括依赖接口(IN_INTERFACE)和提供接口(OUT_INTERFACE)。

ATTRIBUTION_SET 属性集列出仿真构件的属性,主要用来对构件进行参数设置。

OPERATION_SET ::= \langle FUNCTION_NAME, PARAMETER_LIST, RETURN_TYPE \rangle

操作集合通常是一组方法的集合,通常用函数的形式来表示,包括函数名、参数列表、返回类型等。

定义5(构件关系, COMPONENT_RELATION) 描述仿真构件与其他构件的关系,包括构件关系名、相关构件和关于构件关系的详细描述。在构件关系的详细描述中,将具体

介绍这两个构件是如何联系在一起的。

$COMPONENT_RELATION ::= \langle RELATION_TYPE, RELATED_COMPONENT, RELATION_DETAIL \rangle$

构件关系是指构件和其他构件之间的联系,主要有以下几种:

(1)协作关系(Collaboration):指通过与其他构件相互协作共同完成任务,构件之间的协作有依赖关系、供给关系等多种方式。如地形或树木的三维模型构件要依赖于纹理映射绘制构件来映射纹理细节,并通过 OpenGL 库的绘制函数进行绘制。

(2)版本关系(Version):通常指由同一个模型算法构件所演化出的一系列构件之间的关系。如 L 系统派生出的随机 L 系统、开放式 L 系统、参数化 L 系统等,这些模型构件的本质都存在一定的共性。

(3)组合关系(Composition):主要是指构件之间的相互包含关系,实际上包括组合和聚合两种关系,指小粒度构件通过组合形成大粒度构件。

定义 6 (构件实现, $COMPONENT_IMPLEMENTATION$) 主要描述仿真构件实现的相关内容,包括仿真构件形式、开发语言、开发环境、实现技术细节和构件实体资源等。

$COMPONENT_IMPLEMENTATION ::= \langle COMPONENT_FORM, DEVELOPMENT_LANGUAGE, DEVELOPMENT_ENVIRONMENT, IMPLEMENTATION_DETAIL, COMPONENT_ENTITY \rangle$

构件形式 $COMPONENT_FORM ::= \langle DOCUMENT \rangle | \langle SOURCECODE \rangle | \langle BINARY_CODE \rangle | \langle EXECUTABLE_CODE \rangle | \langle PROJECT \rangle | \dots$

森林仿真的研究由于经常是对一些经典模型的优化,因此提供了仿真构件实现的具体描述,包括具体的算法结构等,以便用户能根据该描述对一些模型构件进行个性化的修改。构件资源实体是构件的具体实现部分,是完成一定业务功能模块的软件封装体,用户可以上传、下载或在组装过程动态载入这些构件实体。

在森林仿真服务构件模型的设计上,一般是先满足其强表达能力的要求以便能充分描述构件,然后考虑尽可能简单直观,从而利于对仿真构件的理解和使用,为后期的构件实现和组装提供指导原则。

2.2 仿真构件接口

构件接口是对构件所提供服务的抽象描述,提供了访问构件服务的接口。构件接口规格说明作为构件与外界的一种“通信契约”,必须要能描述构件所实现的功能。构件的接口和构件实现通常被分开,这是基于构件来实现“可装配、可替换、可组合”森林仿真的基础。

森林仿真系统由仿真构件组合而成,因此每个构件需实现一个或多个接口,每个接口通常定义了若干接口函数。构件接口描述为 $COMPONENT_INTERFACE ::= \langle INTERFACE_NAME, INTERFACE_TYPE, ATTRIBUTION_SET, OPERATION_SET \rangle$,接口类型通常包括依赖接口(IN_INTERFACE)和提供接口(OUT_INTERFACE)。

依赖接口(IN_INTERFACE),也称为请求接口或入接口,即描述构件 A 为完成其自身功能接受外界或构件 B 所提供服务的接口;

提供接口(OUT_INTERFACE),也称为供应接口或出接

口,即构件所能提供的服务或数据的输出接口。

如图 2 所示,森林仿真构件与外界的交互主要存在 3 种形式:构件与构件之间、构件与组装框架之间、构件与用户之间。构件的接口将作为桥梁提供连接机制。用户和构件之间交互主要是一些参数的设定及信息反馈,接口所传递的信息通常是属性值;构件和组装框架之间的交互通常是指组装框架在仿真流程的不同阶段,将符合该阶段业务功能需求的构件装配到框架中,通常是指业务构件与整个仿真应用的关系。

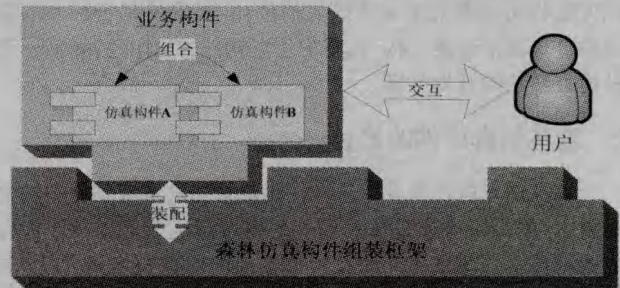


图 2 森林仿真构件参与的交互

构件接口的定义通常为两个具有关联关系的构件进行交互提供了通信规则, $COMPONENT_RELATION ::= \langle RELATION_TYPE, RELATED_COMPONENT, RELATION_DETAIL \rangle$ 。如图 2 所示,仿真构件 CompA 的 $INTERFACE_TYPE = OUT_INTERFACE$,仿真构件 CompB 的 $INTERFACE_TYPE = IN_INTERFACE$,且 CompA 和 CompB 的 $COMPONENT_RELATION = Collaboration$,即两者是协作关系,CompA 提供的值是 CompB 所需要的值,则两者通过接口来传递数据或服务。

下面将以三维地形的生成过程来讲述构件接口的使用。地形建模使用三角网格模型构件 CompTerrainModel,绘制采用纹理映射构件 CompTextureRendering,两者通过接口来协作完成地形的建模可视化。图 3(a)是由 CompTerrainModel 构件所生成的地形网格,而地形的最终生成需要经过绘制构件 CompTextureRendering 来处理。



图 3 三维地形生成过程

三角网格地形构件 CompTerrainModel 的输出接口 OUT_INTERFACE 提供了该模型三角网格数据;绘制构件 CompTextureRendering 的输入接口 IN_INTERFACE 通过接口函数 Get_the_terrian_data() 获取 CompTerrainModel 输出接口所能提供的参数值,并结合图 3(b)的纹理图片,来完成地形的映射和绘制,生成如图 3(c)所示的三维地形。

由于森林仿真涉及到多种交互,本文在设计仿真构件接口时遵循两条准则:(1)应尽可能地灵活,以支持互操作;(2)充足的接口,可以预留一些未被实现的接口供扩展功能使用。用于构件组装的框架上设计的每个接口通常有多种实现,可以在不断的变化中随需应变,因为基于该框架的业务构件通常可以由不同的仿真构件组合而成,如森林场景地形的生成,可以采用不同的模型和不同的纹理实现,因此组装框架应当提供较详尽的构件交互规则,并基于这些规则实现类似容器的标准环境。接口使得设计与实现分开,构件组装者不需要过多关心构件的内部实现,但构件接口的描述(表达能力和完

整性)应该足够完备,因为这将是构件使用者依赖的信息来源。

2.3 仿真构件封装

森林仿真构件的实现借鉴面向对象(Object-oriented)的优秀特点:封装和聚合,即对森林仿真中的业务逻辑进行封装,业务构件的实现通常是通过仿真构件的组合来完成,因此本文主要是针对仿真构件进行封装实现。森林仿真中,最具可替换性的是各种仿真模型。因此,本文通过提取出的地形模型、天空模型、树木个体生长模型、植物间相互作用模型、森林动态演变等模型进行抽象归纳,提取特征参数,基于灵活的接口的设计,封装这些模型的内部逻辑,甚至将一些可视化算法分解成恰当规模和结构的模块并封装成构件。

考虑到基于组装框架完成构件组装,需要动态插入或替换仿真构件,因此仿真构件实体最好不重新编译,也不局限于某种语言,它尽可能采用二进制标准来封装^[15]。编译之后的构件以二进制形式发布,可以独立部署,又能在集成环境下运行,也就是实现基于接口的构件复用,而不是源代码级别的复用。通常要对源码进行编译打包,封装成如下几种常见的二进制形式:

〈二进制|BIN〉::=〈DLL〉|〈LIB〉|〈COM|DCOM〉|〈ActiveX〉|〈CORBA〉|〈JAR〉|〈OTHER〉

静态库 LIB 在编译时调用;动态 DLL 可以定义要公开的函数名称,在运行时调用;CORBA 需要使用 IDL 定义接口,使用 IDL 编译器得到 Stub 和 Skeleton,然后使用相应语言的编译器来实现 CORBA 构件;JAR 包是与平台无关的文件格式,但需要 JVM 的支持。

构件并非一定要用面向对象语言来编写,任何一种可以实现构件标准接口和所需功能的语言都可以用来编写构件。构件组装时只需要按照构件的标准接口来实现构件。通常由于面向对象语言针对具有复杂逻辑的业务可以清晰地定义接口,而且具有封装性、继承等特点,能有力地支持复用,因此应尽可能考虑采用面向对象方法来封装。

3 业务流程驱动下的森林仿真构件组装方法

在森林仿真构件组装过程中,需要一个支撑框架来组织这些仿真构件,并指导这些构件的交互和协作从而完成森林仿真应用的构建。构件组装目前仍然缺乏通用的工具与方法,本文采用多种构件协作方式结合的、以框架为主导的构件组装策略。仿真构件组装框架作为森林仿真构件组装的蓝图,主要解决构件互操作问题,是一些相互合作的类的集合,包含了仿真流程以及一些设计决策。仿真构件基于框架的指导发生交互,从而支持各种独立构件进行有效的协作,并最终能组装得到森林仿真应用。

3.1 仿真构件组装框架设计

森林仿真构件组装框架是构件相互协作以及构件组装的支撑结构。该平台可以理解是一种大粒度的、抽象级别较高的构件,是对仿真系统架构中表示层和数据层的封装,它为仿真构件的组装提供了基础和上下文。本文按照森林仿真的核心业务流程,从仿真系统中分解出了几个重要的业务模块:地形模块、天空模块、植物模块、特效模块等,并将这些业务构件剥离出来,形成一种“插槽式”的框架,如图 4 所示。这些业务模块作为可替换的构件接入框架中,并基于业务流程驱动的总线型框架来完成协作与组装工作。

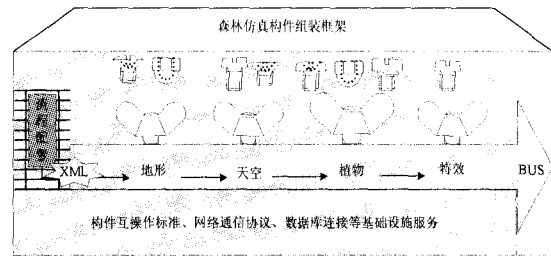


图 4 森林仿真构件组装框架概图

森林仿真构件组装框架的设计基于产品线、框架技术、业务流程驱动、服务总线等思想,目的是提供一个通用的、可定制的森林仿真快速组装平台,并提供一系列的构件接入点为构件组装提供支持,用户可以根据不同的仿真需求通过选配合适的仿真构件来快速集成森林仿真应用。基于产品线的架构在突出不变的同时识别允许的变化,并提供装载变化的机制。通常将架构用框架的形式予以实现,形成一个平台,供后期产品生产的定制使用。框架技术本质上是将系统的通用部分和专用部分分割开来,由变化较小的通用部分来构建系统的主体,将易变的专用部分封装成扩展点,由框架上的一系列扩展点来接受专用部分的填充。这样的组装框架带来了更好的易修改性和可重用性。

森林仿真构件组装框架提供构件连接和交互协议的严格定义,并基于数据总线来共享核心数据,控制仿真流程等。本文参考体系结构描述语言 ADL^[16],将基于组装框架的森林仿真系统架构描述为:系统架构={构件 Component、连接器 Connector,配置 Configuration}。

Component:从森林仿真系统中剥离出来的一些核心模块,完成每个业务模块的可以是一个单独的构件,通常是由多个小粒度的仿真构件组成。

Connector:在框架平台上针对业务构件设立“插槽式”接口,是基于“策略模式”思想,即针对同一问题往往存在一系列可以相互替换的解决方案。本文将每种方案封装成构件,这些构件具有相同的接口或经过适配后可以使用,从而使得这些森林仿真构件可以相互替换。这是由于森林场景的地形、树木往往都有多种建模可视化方法,因此可以将这些技术方法封装成构件,在组装过程中按需装配“即插即用”,从而可以快速响应森林仿真领域的个性化需求,实现仿真应用的高度“可定制”和“可配置”。

Configuration:框架平台中的服务总线是基于森林仿真业务流程驱动,提供构件接口机制和交互规则,并保证不同模块之间的衔接。该框架通过配置流程和制定构件交互的一些规则,实现不同层次构件之间的数据传递,从而可以保证组装的流程化进行以及构件与底层系统的交互。平台的底层基础设施提供构件互操作的标准、通信协议、数据库连接等支撑。

该组装框架的设计体现了开放性、可配置性、可插拔性、可扩展性等优点。森林仿真应用的搭建通过基于业务流程驱动的构件的协作和装配来完成,使得森林仿真研究的关注点将不再是具体仿真算法技术的实现,而是在于如何提取并设计灵活的构件,并利用这些仿真构件来实现快速装配。

3.2 业务流程驱动的仿真构件组装

在森林仿真过程中,不同仿真系统的仿真流程非常相似,而构建每个场景实体时所采用的模型、算法或技术方法往往却有很多种。因此,本文通过分析森林生长仿真的业务流程,

并将其作为贯彻森林构件组装框架的主线,其构件化的森林仿真流程如图 5 所示。在仿真流程的不同阶段,根据应用目标的不同,可以选择不同的构件。地形生成阶段可以选择三角网络模型构件、规则网格模型构件或者采用不同的分形技术建模构件;天空生成阶段可以选择半球顶模型构件或天空盒模型构件等;树木生成时可以选择不同的形态模型构件以及不同的生长模型构件。因此,通过在仿真流程的不同阶段选择不同的构件,可以快速搭建面向不同应用目标的森林场景仿真。

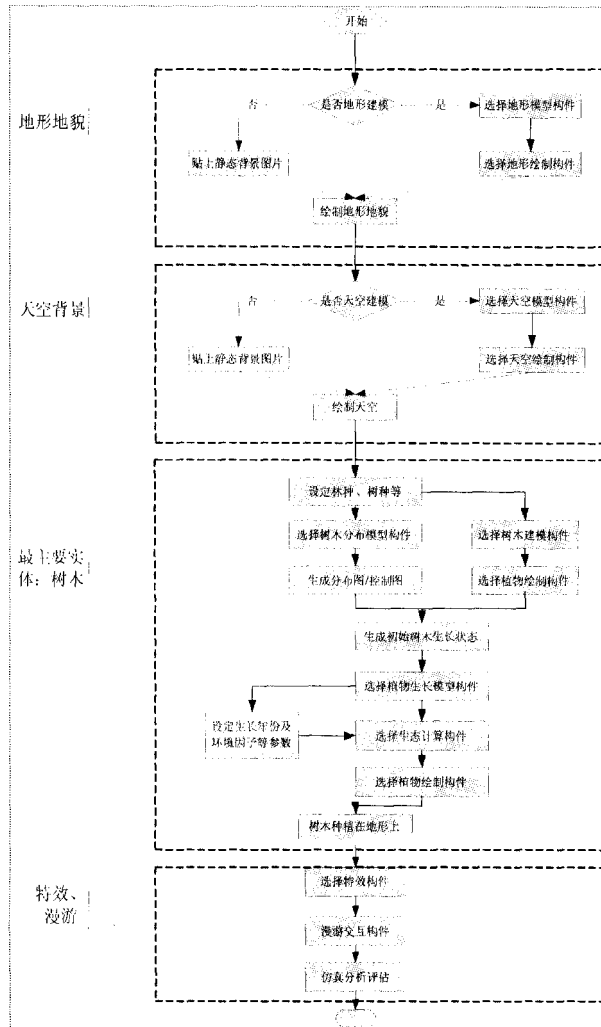


图 5 基于构件组装的森林仿真流程

基于构件组装框架的森林仿真以选取仿真构件来装配为主,本文利用构件模型来描述基于仿真流程驱动的构件装配过程。表 1 为组装过程中涉及到的主要构件模型变量和参数,这些参数用来描述基于业务流程驱动的构件化森林场景组装过程。图 6 和图 7 详细描述了基于框架的森林仿真构件组装过程,该组装过程将使用表 1 所列的参数。在整个仿真过程中,地形 CompTerrain、天空 CompSky、树木 CompTree 可以作为相对独立的业务构件来分别生成,而这些业务构件往往是通过仿真构件的组合来完成。这 3 个业务构件基于组装框架通过彼此协作来完成整个森林场景的构建,如地形 CompTerrain 和天空 CompSky 通过彼此的边界值来完成协作,防止天空和地形之间出现裂痕等;树木 CompTree 和天空 CompSky 的协作要考虑树木应置于天空前段,并且保证树木不能种植于天空上;树木 CompTree 与地形 CompTerrain 将

通过树木根部坐标值与地形网格点坐标值的匹配来实现交互,防止出现树木埋入过深或漂浮在空中的现象。

表 1 构件组装过程涉及的部分构件模型参数

属性	子属性	取值	功能描述
SIMULATION_CONTEXT		Terrain_Modeling	用于建立地形三维模型
		Sky_Modeling	用于建立天空三维模型
	BUSINESS_FUNCTION	Tree_Modeling (Tree_Morphology_Model, Tree_Growth_Model)	用于植物建模,又具体分为形态、生长模型
	Calculation Rendering		用于生态计算等 用于绘制渲染
POST_CON_STATUS		0	表示该构件后置条件不满足,不能继续执行
		1	表示该构件后置条件满足,或者不需要后置条件
COMPONENT_INTERFACE	INTERFACE_TYPE	IN_INTERFACE OUT_INTERFACE	输入接口 输出接口
	ATTRIBUTE_SET	{...} ... {...}	属性集合
COMPONENT_RELATION	OPERATION_SET	{FuncA(), FuncBC(), ...}	操作集合
	RELATION_TYPE	Collaboration Version Composition	相互协作关系 版本关系 组合关系
COMPONENT_IMPLEMENTATION	RELATED_COMPONENT		关联构件
	COMPONENT_ENTITY		构件实体 用于组装

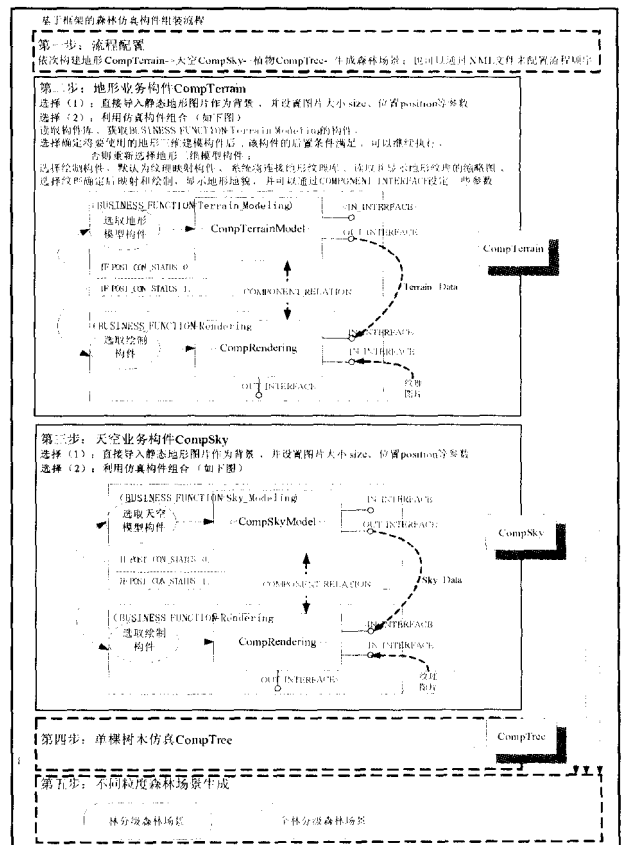


图 6 森林仿真构件组装过程(一)

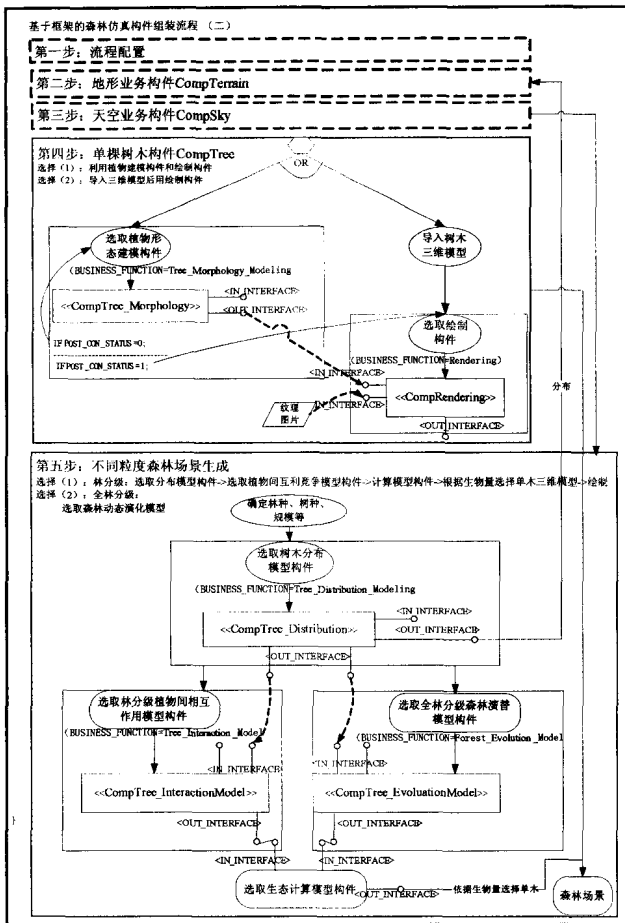


图7 森林仿真构件组装过程(二)

在图6和图7所示的森林仿真构件组装过程中,构件组装时首先将符合森林仿真业务流程的仿真构件从存储构件的数据库中读取并载入,然后将其装配成仿真系统的功能模块。一般尽量将仿真构件绑定成较大的业务构件,再根据构件的接口进行装配。当用户在仿真过程中更换各种模型或算法时,只需替换该模型构件,而不需要牵涉其他模块,基于框架平台的组装将充分支持仿真构件的动态更替。组装构件时,有时必须编写粘接代码,这些代码能够消除构件间接口不兼容的问题,同时为系统提供统一的异常处理机制。

4 基于组装框架快速构建森林仿真应用

4.1 构件化森林仿真的体系结构

本文以构件思想架构森林仿真系统为研究核心,图8展示了基于构件进行森林仿真的整个体系结构。本文设计了基于B/S和C/S架构结合的体系,这种架构支持灵活的构件提交,不同的开发人员可以并行进行构件开发,并且可以在不同地点,通过B/S架构的构件管理工具将仿真构件上传到数据库中;当进行森林仿真时,通过C/S架构的组装框架读取数据库中构件,来快速集成森林仿真应用。

采用B/S架构的构件管理工具来实现仿真构件的提交以及构件存储和管理,构件开发者可以将森林仿真领域可复用的资产按照一定的接口标准和公开的构件规范封装成构件,并存放在有良好目录定义、可控的构件数据库中。B/S架构使构件开发者可以通过浏览器远程提交开发好的构件,以便构件管理者管理这些构件,为基于构件组装来实现森林仿真应用的快速构建提供基本元素。

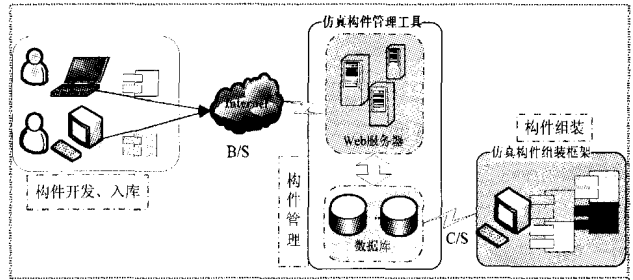


图8 构件化森林仿真的体系结构

仿真构件组装框架突破了以往森林仿真系统的集中式架构,是基于C/S架构来从存储构件的数据库里获得构件,并利用构件组装来完成森林仿真系统的集成。通过组装框架访问数据库,并根据具体需求,在预先定义好流程的框架上,通过读取数据库中符合流程模块需求的仿真构件,来快速构建不同应用目标的森林仿真系统。基于框架的构件组装提高了森林仿真的可管理性和可维护性。

4.2 基于组装框架快速构建森林仿真应用

森林仿真构件组装框架作为构件运行和组装的支撑环境,将通过仿真流程来实施构件的连接。该组装框架是在VC6.0环境下基于MFC和具有强大绘图功能的OpenGL库来实现,并且连接森林仿真构件管理系统的MySQL数据库。

本文通过仿真业务流程驱动组装过程,实现了森林仿真构件组装框架,从而使得在森林仿真过程都是通过选取式操作并输入参数信息来进行。用户可以在不同的仿真流程环节,选择不同的操作来快速响应变化的仿真需求,而不是一切从代码编写开始。用于组装的仿真构件是从构件管理系统的数据库中远程动态读取,构件管理提供了该构件的信息描述以及与关联构件关系的描述。进行构建组装时,可以通过组装框架访问这些构件信息,并基于这些信息来驱动整个仿真的流程。当需要选取构件时,可以发出“读取”命令,向数据库提交检索请求,从构件数据库中检索该仿真流程环节所需的构件,通过读取仿真构件来装配完成流程各个环节的仿真业务。这种基于已有构件的组装,可以更快地集成森林仿真系统。系统提供了可视化界面来对森林仿真系统进行层次清晰的建模可视化和构件组装,用户可以通过界面操作连接构件数据库、配置系统参数、选取仿真构件等。

搭建森林仿真应用时,首先是要连接构件管理系统的构件库,将通过系统中构件库连接设置模块进行配置,并进行仿真构件的管理和维护,如图9所示;然后,按仿真流程,地形构件—>天空构件—>树木构件—>不同粒度森林场景的生成,进行不同的地形、天空等场景的构建,并进一步构建树木和森林仿真实体。

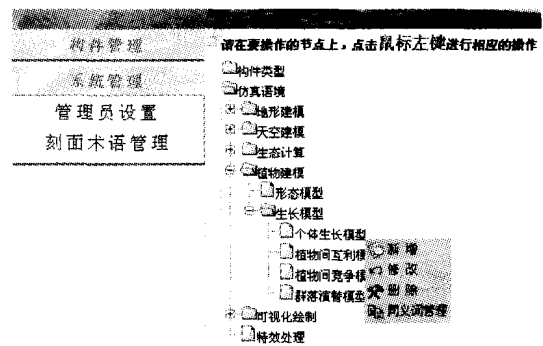


图9 森林仿真构件界面管理

本文在构建森林仿真系统时,选择了纯种林,然后进行参数设置和模型构件的选取,如图 10 所示。在该纯种林设置窗口中,下拉列表显示了已经建好的单木模型的树种。本应用实例选择了落叶松模型,设置数量、树龄等相应的参数,然后读取构件数据库中有林分分布的模型构件,并选择井型分布模型构件,通过分布设置来设置分布位置和间距等。该应用实例中选择 4*5 的分布模型,并布局 20 棵落叶松,分布概图直观显示分布情况,并设定植物间交互的生长模型构件,从而得到整个场景的绘制效果,如图 11 所示。

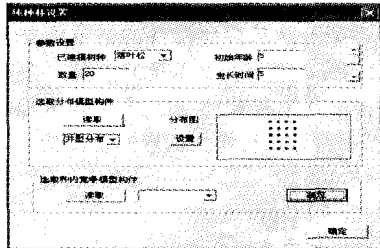


图 10 纯种林设置

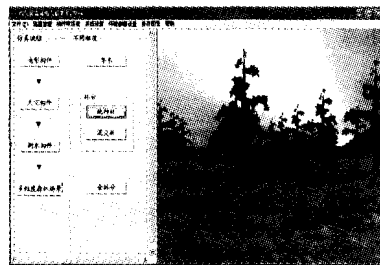


图 11 落叶松纯种林仿真效果

表 2 采用不同方法构建森林仿真系统的工作量估算和比较

仿真系统的构件或模块	构件或模块的功能描述	开发或组装的工作量(人小时)		
		重新开发	代码重用	构件组装
System_Struct	用于创建森林仿真系统的框架,以及系统集成	300	200	77
Terrain_Modeling	用于建立地形三维模型,包括草地、雪地、沙漠等地形场景	150	5	2
Sky_Modeling	用于建立天空三维模型,包括晴天、阴天、雨天、雾天、黄昏等情况下的天空场景	150	5	2
Tree_Distributing_Model	用于建立森林场景中树木的分布模型,包括井型分布、均匀分布、随机分布等	200	10	5
Tree_Morphology_Model	用于三维植物模型的处理,包括模型的读取、编辑和存储	200	11	5
Tree_Growth_Model	森林生长仿真模型的表达,包括林窗模型、Lotka-Volterra 模型等的表示与存储	200	22	10
Calculation	用于森林生长仿真模型的计算	300	40	11
Rendering	用于绘制渲染,能根据视点进行不同分辨率森林场景的绘制、漫游等操作	250	26	10
开发总时间(人小时)		1750	319	122

基于该仿真构件组装框架,使森林场景的各个组成部分都支持独立的用户修改。当用户的仿真需求有所改变时,只需重新选取适合其需求的仿真构件,而不需要对整个系统的实现进行重构,便能支持不同仿真粒度森林场景的快速构建,这有利于用户直观比较基于不同构件实现的森林场景效果。

为了说明采用构件组装方法进行森林仿真系统快速开发的优势,本文对采用构件组装、构件的代码级重用和重新开发 3 种方式进行森林仿真系统开发的工作量估算和比较,如表 2 所列。其中,采用重新开发方式时,需要完成森林仿真系统的框架、各个模块或构件的设计和代码编写,大概需要花 1750 小时每人左右来完成整个系统的开发;在代码级重用方式中,对系统的架构以及各个模块或构件的设计和代码进行重用,完成系统开发大概需要 319 人小时;而采用构件组装方式,完成森林仿真系统的构建大约只需要 122 小时。

当森林仿真系统的部分需求发生变化时,采用构件组装方式的优势更为明显。比如,要将森林仿真系统中原来使用的林窗模型替换为 Lotka-Volterra 模型时,需要重新开发 Tree_Growth_Model 和 Calculation 两个构件,大概需要 500 人小时,而采用构件组装方式时只需要 21 人小时。需要说明的是,所花费的 21 人小时中有很大一部分工作是用于了解构件的功能和使用方法,真正的构件组装和调试只需要几个小时而已。

结束语 针对森林仿真应用系统的快速搭建和重构的问题,提出了业务流程驱动的森林仿真构件组装方法,并将其用于虚拟森林仿真系统中。本文面向森林仿真构件模型提出了构件接口描述与构件封装方法,并设计了森林仿真构件组装框架。该组装框架是基于仿真流程驱动的“插槽式”平台,为构件组装提供上下文环境,使得森林仿真主要环节的模型算法可以动态替换,具有良好的可扩展性和可维护性。本文还按照仿真流程驱动的构件装配过程快速构建了森林仿真应用系统。该应用表明,利用业务流程驱动的森林仿真构件组装方法,可以有效地提高森林仿真软件的复用性,降低了森林仿真系统开发的难度。

参考文献

- [1] Uusitalo J, Orland B. Virtual Forest Management: Possibilities and Challenges [J]. International Journal of Forest Engineering, 2001, 12(2): 57-66
- [2] 范菁,孙思昂,董天阳. 面向森林动态生长过程的场景系统设计和实现[J]. 计算机应用研究, 2008, 25(9): 2872-2874
- [3] 徐丙立,龚建华,林琛. 基于 HLA 的分布式虚拟地理环境系统框架研究[J]. 武汉大学学报:信息科学版, 2005, 30(12): 1096-1099
- [4] 孙思昂. 基于 HLA 的动态森林生长仿真原型系统研究与设计[D]. 杭州:浙江工业大学, 2007
- [5] 唐丽玉,陈崇成,王钦敏,等. 基于 HLA 的虚拟森林环境构建[C]//第六届全国地图学与 GIS 学术会议. 2006
- [6] 段峥嵘,陆守一. 组件式 GIS 及其在森林资源信息管理中的应用[J]. 林业资源管理, 2003(3): 50-53
- [7] 叶劲松,陆守一,孙美娟,等. 基于组件的网络化森林资源信息集成系统的研建[J]. 农业网络信息, 2005(6): 36-38
- [8] 丁维龙,熊范纶,张友华. 基于构件的植物三维结构模拟模型[J]. 小型微型计算机系统, 2004, 25(9): 1624-1627

2) Web 服务集的产生

Web 服务集来自于真实的服务网站的 WSDL 文件,如 webxml 网站^[8],是真实的数据集,共采用 200 个服务进行实验分析。

3) 服务访问日志的产生

目前没有相关的 Web 服务访问日志的标准平台和标准测试数据集。本文采用模拟生成日志作为测试日志,即模拟 100 个用户随机访问服务,使其产生 2000 条服务访问日志。此外,将实验中产生的服务访问记录继续添加到实验用日志中。随着访问次数的逐渐增加,服务访问日志里的日志项也随之增加,从而使得依据访问日志所得的分析结果趋向普遍性。

4.2 衡量标准

本文采用准确率、召回率和匹配时间 3 个度量标准来衡量基于服务访问日志的服务发现方法和一般的服务发现方法。准确率和召回率的常规计算公式为:准确率=(检索出的相关正确的服务数/检索出的服务总数) * 100%;召回率=(检索出的相关正确的服务数/正确的服务总数) * 100%;

4.3 实验结果分析

本文通过对基于关键字的服务发现方法和基于服务访问日志的服务发现方法的结果进行综合比较,得到初步对比结果,如图 3—图 5 所示。

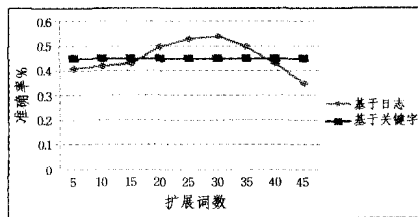


图 3 两种方法的准确率比较

由图 3 可知,基于服务访问日志的服务发现方式的准确率是否高于基于关键字的服务发现方式的关键是扩展词的选择。当扩展词数在 25~35 之间时,本文提出的服务发现方法的准确率明显高于基于关键字的方法。实验表明,在查询词中加入 20~30 个扩展词,查询性能达到更高,超过 30 个后查询性能将下降得很快。

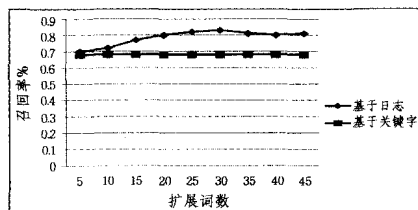


图 4 两种方法的召回率比较

由图 4 可知,基于服务访问日志的服务发现方式在召回

率上明显高于基于关键字的服务发现方式。

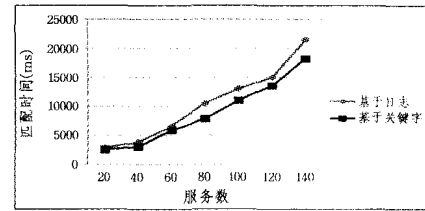


图 5 两种方法的匹配时间比较

由图 5 可知,基于不同的服务数进行服务匹配,本文提出的服务发现方法花费了较多时间。本文所提出的方法是建立在大量用户长期查询访问服务行为的基础上的,它在选择扩展词汇时更具有针对性,避免扩展用词扩散,不影响后续查询。

结束语 高效的服务发现方法是服务应用的关键。以往的研究主要集中在语义描述上,本文通过挖掘服务访问日志,建立查询词之间以及查询词和服务之间的联系,从而在保证服务发现质量的前提下,提高服务发现的准确率和召回率。本文提出的方法中,如何更好地确定查询词之间的关系,取消扩展词的歧义以及如何提高查询效率,将是下一步的研究重点。

参考文献

- [1] Liang A, Miller S, Chung J Y. Service mining for Web service composition[C]//Proc of IEEE International Conference on Information Reuse and Integration, Las Vegas; s. nl, 2005; 470-475
- [2] 朱红康,余雪丽. 基于用户和服务协同聚类的 Web 服务发现研究[J]. 计算机应用研究, 2010, 27(3): 986-988
- [3] Wen J R, Nie J Y, Zhang H J. Clustering user queries of a search engine[C]//Proceedings of the 10th International World Wide Web Conference (WWW10). New York: ACM Press, 2001: 162-168
- [4] Paolucci M, Kawamura T, Payne T, et al. Semantic matching of Web services capabilities[C]//Proc of the 1st International Semantic Web Conference. London, UK: Springer-Verlag, 2002: 333-347
- [5] Syeda-Mahmood T, Shah G, Akkiraju R, et al. Searching service repositories by combining semantic and ontological matching[C] // Proc of IEEE International Conference on Web Services. Washington DC: IEEE Computer Society, 2005; 13-20
- [6] 崔航,文继荣,李敏强. 基于用户日志的查询扩展统计模型[J]. 软件学报, 2003, 14(9)
- [7] Wu Jian, Wu Zhao-hui. Similarity-based Web service matching [C]//Proceedings of the 2005 IEEE International Conference on Service Computing (SCC'05). IEEE Computer Society press, Orlando, Florida, USA, 2005: 287- 294
- [8] Web Service(Web 服务)[BP/OL]. <http://www.webxml.com>

(上接第 132 页)

- [9] 丁维龙,熊范纶. 植物模拟组件的设计与实现[J]. 中国科学技术大学学报, 2003, 33(6): 742-748
- [10] 姜海燕,朱艳,曹卫星,等. 作物模型资源构造平台(CMRCP)的构建研究[J]. 农业工程学报, 2008, 24(2): 170-175
- [11] 王忠芝,胡逊之. 基于 Xfrog 的树木建模及生长模拟[J]. 北京林业大学学报, 2009, 31(增刊 2): 64-68
- [12] 赵海燕,张伟,麻志毅. 面向复用的需求建模[M]. 北京:清华大学出版社, 2008

- [13] 黄翌,张路,周明伟,等. 构件化软件设计与实现[M]. 北京:清华大学出版社, 2008
- [14] 谢冰,王亚沙,李戈,等. 面向复用的软件资产与过程管理[M]. 北京:清华大学出版社, 2008
- [15] 计算机软件构件复用属性规范[EB/OL]. <http://www.sawin.cn/doc/Document/DocCase/blueskil126.htm>, 2010-12-23
- [16] 王晓光,冯耀东,梅宏. ABC/ADL: 一种基于 XML 的软件体系结构描述语言[J]. 计算机研究与发展, 2004, 41(9): 1521-1531