一种基于功能构件的Web应用建模与测试方法

唐云吉[1] 缪淮扣[1] 钱忠胜[1,2]
(上海大学计算机工程与科学学院 上海 200072)[1] (江西财经大学信息管理学院 南昌 330013)[2]

摘 要 Web应用与传统程序有着很大差别,后者的一些建模和测试方法不能完全适用于前者。提出了一种有效的对Web应用测试的方法。按功能将Web应用划分成若干个功能构件,把Web应用看成是功能构件的集合,并在功能上将其对应到实际的Web应用模块。用有向图表示功能构件的结构关系,用FSM表示功能构件的行为关系,用FSM的复合表示功能构件的交互。提出了完整执行序列覆盖、构件完整执行序列覆盖两个测试准则,根据这些准则生成测试用例集。为支持所提出的方法,设计了一个测试用例生成的工具原型。

关键词 Web应用,功能构件,建模,测试准则,测试用例
中图法分类号 TP311 **文献标识码** A

Approach to Modeling and Testing Web Applications Based on Functional Components

TANG Yun-ji[1] MIAO Huai-kou[1] QIAN Zhong-sheng[1,2]
(School of Computer Engineering & Science, Shanghai University, Shanghai 200072, China)[1]
(School of Information Technology, Jiangxi University of Finance & Economics, Nanchang 330013, China)[2]

Abstract There are great differences between Web applications and the traditional programs. The modeling and testing methods of the traditional programs can not be fully fit for Web applications. An approach to generating test cases effectively was presented. A Web application was divided into a set of functional components, each of which corresponded to an actual Web application module. A directed graph is employed to represent the structural relationship among the functional components, an FSM was used to represent their behavioral relationship, and the FSM composition was used to represent their interactions. In testing, two test criteria which are complete executing sequence coverage and component complete executing sequence coverage were proposed. To satisfy these two criteria respectively, different test sets were generated. Additionally, a test case generation prototype was designed for the approach proposed.

Keywords Web application, Functional component, Modeling, Test criterion, Test case

1 引言

随着网络互联技术、信息技术以及Web技术的发展,存在越来越多的基于Web的应用,在这些应用中,人们可以方便快捷地处理各种各样的信息。然而,Web应用由不同种类的软件构件组成,它们之间以及和用户之间以全新的方式进行交互。软件构件必须满足高的可靠性、可用性和有效性需求。Web应用的建模、测试和分析对软件开发者和设计者提出了新的挑战。

Web应用有别于传统的软件系统,对其测试带来了许多新的问题[1,2]。构建一个即使是简单的Web应用也是一项很耗时的任务,我们必须对其结构和行为建模并仔细分析。Web测试方法也需自动化,测试用例也要有自适应能力。Web应用提出了一些重要而具有挑战性的测试问题,这些问题不能直接用现有的对传统系统的测试技术来解决。例如,Web应用的Session控制、Cookies控制和一些新的安全问题等都是传统系统所没有的。本文从Web应用的功能需求出发,提出一个构件建模的方法对Web应用进行建模,然后根据提出的测试准则自动产生抽象的测试用例。

2 相关研究

Web软件的主要特征是封装性、松散耦合性、跨组织和跨平台性。这些特征使得传统的软件测试方法和技术很难实现对Web应用进行系统的测试。目前针对Web软件的测试模型、测试策略、测试层次和测试过程方面的研究工作刚刚起步。Brim[3]提出了基于构件交互自动机的方法对构件交互行为进行建模。Andrews等[4]分析构成Web应用的网页和软件构件之间的8种连接关系,提出了一种基于FSM的Web应用建模和测试用例生成方法。该方法通过对Web应用进行功能簇和逻辑网页的划分并用带约束的分层FSM来表示逻辑网页及其导航关系,但没有进一步考虑Web应用中软件构件的交互性和合成的测试问题。文献[5]提出了一种基于

到稿日期:2009-01-16 返修日期:2009-03-25 本文受国家自然科学基金项目(60673115),国家863计划项目(2007AA01Z144),国家973项目(2007CB310800),上海市教委科研项目(07ZZ06),上海市重点学科建设项目(J50103)资助。
唐云吉(1984—),男,硕士生,主要研究方向为Web应用的建模与测试,E-mail:yunjitang@shu.edu.cn;缪淮扣(1953—),男,教授,博士生导师;钱忠胜(1977—),男,博士生,讲师。

逻辑构件的对 Web 应用进行划分的方法，并重点考虑了 Web 构件的交互问题，对构件交互的测试用例的自动生成进行了深入研究，同时利用 Agent 来处理复合中的不匹配动作。文献[6]提出对 Web 应用进行逻辑构件划分，提出了单结点测试准则，对 Web 应用的交互性生成测试用例。D. C. Kung 和 Chien-Hung Liu 等[7,8]提出了一种基于包括对象关系图、对象状态图、脚本簇图和网页导航图等多模型在内的测试用例生成方法，但这一方法在软件对象很多、状态空间非常复杂的时候难以奏效。总的说来，现有的 Web 软件的测试研究只是分别考虑 Web 软件测试的一个或几个方面，没有关注面向 Web 软件整体的建模以及基于 Web 软件的模型产生测试用例的方法，这方面的研究还是比较初步的。

3 Web 应用的建模

3.1 基本概念

在 Web 应用程序中，构件可以抽象理解为是 Java applet，ActiveX 控件，Java Bean，HTML 模板文件或用任意编程语言编写的程序或程序集合。一个 Web 应用程序可以看成是一个构件库的集合，构件之间的交互及相互作用，引发了 Web 应用程序的多姿多态，使得 Web 应用程序具有可交互性，易操作性。每个构件就像一个个细胞，相互分工，相互协作地工作。从总体上来看，Web 应用程序的测试关注的焦点就是这些细胞单元之间的关系，我们考虑的前提是这些 Web 构件是经过充分性测试的。下面给出功能构件和构件连接器这两个概念。

功能构件。一个整体封装好的结构单元，能独立完成 Web 程序中的功能。从用户视角看，Web 应用是一个黑盒，可按照功能将 Web 应用细化成构件。功能构件由功能构件名和两类接口（输入接口、输出接口）组成。功能构件通过不同的动作向输出接口发送数据或从输入接口接收数据。一个功能构件又可以是若干个功能构件的集合。一个功能构件可以由多个粒度更细的功能构件组合而成。例如一个信息管理模块，可以看成是一个功能构件，再细分一下，这个功能构件可以分为“查询”功能构件、“编辑”功能构件等。至于划分的粒度，根据测试需求来定义。

构件连接器。一个连接器是对构件之间的通讯、协调或者合作进行仲裁的一种抽象机制，它是构件与构件之间进行通信、协作的桥梁。

3.2 功能构件的划分与建模

本文以一个小型新闻发布（News Publishing）Web 应用为例展开研究，根据该 Web 应用的不同功能划分功能构件。图 1 是它的构件依赖图，构件之间的依赖特性由构件连接器表示，每一个箭头表明，如果这个构件要触发事件，就需要指定它所依赖的构件先执行。例如，如果构件 PublishNews 要正常工作，就必须先要运行 Login；在构件 PublishNews 中，如果想执行构件 EditNews，就要先执行构件 NewsEditor，而一个 PublishNews 构件的完成则需要构件 NewsEditor 和 EditorNews 协同完成。

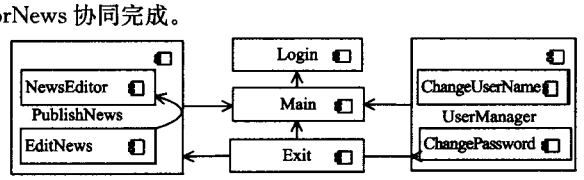


图 1 构件依赖图

可以定义该小型新闻发布 Web 应用为一个构件集合，最高层集合 WebSet = {PublishNews, UserManager, Login, Main, Exit}；而 UserManagerSet 是构件 UserManager 的集合，UserManagerSet = {ChangePassword, ChangeUserName}；PublishNewsSet 是 PublishNews 构件的集合，PublishNewsSet = {EditNews, NewsEditor}。

构件之间的关系可用一张构件关系表 $K=\{V,L\}$ 来表示。其中 V 是所有构件的集合，表示划分过程中的所有功能构件。$L=\{T_1,T_2,T_3\}$ 是功能构件关系种类的集合，其中 T_1,T_2,T_3 均为构件对组成的集合，即有 $T_1,T_2,T_3:V\leftrightarrow V$，并且

- $(x,y)\in T_1$ iff $x\in y$；即对 T_1 中的任意一个序偶 (x,y)，x 属于 y。
- $(x,y)\in T_2$ iff $\exists z\in V\cdot x\in z\wedge y\in z$；即对 T_2 中的任意一个序偶 (x,y)，x 和 y 同属于同一个功能构件。
- $(x,y)\in T_3$ iff $(x,y)\notin T_1\wedge(x,y)\notin T_2$；即 T_3 表示构件间除 T_1,T_2 之外的其它关系。

表 1 能清楚地描述新闻发布 Web 应用中各个构件之间的关系，这可以用来指导测试用例的生成。根据给出的构件依赖图与构件关系表，可以用有限状态机构造出它的形式测试模型，如图 2 所示。

表 1 构件关系表

构件 A	构件 B	类型	是否交互
Login	Main	T3	No
Login	Exit	T3	No
ChangePassword	ChangeUserName	T2	No
NewsEditor	EditNews	T2	Yes
PublishNews	UserManager	T3	No
EidtNews	NewsEditor	T2	Yes
……	……	……	……

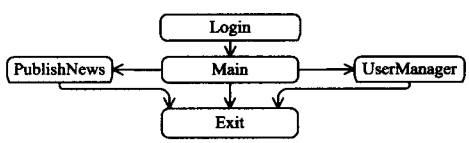


图 2 构件依赖图的顶层 FSM 模型

图 3 和图 4 分别是 PublishNews 和 UserManager 的内部构件的 FSM 模型。这两个模型有些不同，因为 PublishNews 的内部构件有依赖关系，如果要完成 PublishNews 构件的执行，必须先执行 NewsEditor，然后执行 EditNews。UserManager 里面的构件无依赖关系，ChangeUserName 与 ChangePassword 可以任意执行其一。这两种情况都认为构件与构件之间发生了交互。

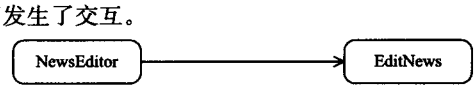


图 3 PublishNews 内部构件 FSM 模型

图 4 UserManager 内部构件 FSM 模型

3.3 构件交互的建模

Web 应用的行为一般很复杂，因为构件之间的交互行为会由于浏览器的前进（forward）、后退（back）、刷新（refresh）、URL 重写（URL rewriting）等功能而变得复杂。为了降低模型的复杂度、减少生成测试用例的数量，指定构件之间需要进行交互测试。

构件的行为及其交互行为可以用 FSM 复合机来描述，它是一个五元组(Q,ACT,δ,I,S)，其中，

- Q 是有限状态集；
- Act 是有限动作集，$\Sigma=((X\cup\{-\})\times ACT\times(X\cup\{-\}))\backslash(\{-\}\times ACT\times\{-\})$，其中 $X=\{c\mid c$ 是功能构件，c 出现在 S 中$\}$；符号“－”表示没有指定功能构件；
- $\delta\subseteq Q\times\Sigma\times Q$ 是状态迁移集；
- $I\subseteq Q$ 是非空初始状态集；
- S 是由功能构件名构成的多元组。

对于功能构件 A 和 B，用$(A,a,-),(-,a,B),(A,a,B)\in\Sigma$ 表示 A 和 B 的交互。输入符号$(-,a,B)$代表构件 B 接收一个动作 a 作为输入；输出符号$(A,a,-)$代表构件 A 触发一个动作 a 作为输出；内部符号(A,a,B)表示 A 发出一个动作 a 作为输出，而 B 接收 A 发出的动作 a 作为输入。

根据图 5 的构造方法以及表 2 的构件动作表(注：构件的每个动作都对应为一个实际的操作)，可以给出一些构件交互 FSM 的实例如下：

- 仅有输出接口的构件 NewsEditor 对应的有限状态机为：FSM_NewsEditor ＝({q0},{a013},{(q0,(NewsEditor,a013,－),q0) },{q0},{NewsEditor})；
- 仅有输入接口的构件 Login 的有限状态机为：FSM_Login ＝({q0},{a001},{(q0,(－,a007,Login),q0)},{q0},{Login})；
- 具有输入接口及输出接口的构件 EditNews 的有限状态机为：FSM_EditNews ＝({q0,q1},{a014,a015},{(q0,(－,a014,EditNews),q0),(q0,(EditNews,a015,－),q1)},{q0},{EditNews})。

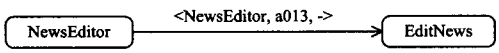


图 5　构件交互的 FSM 复合机

表 2　构件动作表

构件名	构件 ID	构件动作
Login	1	⟨Login,a001,－⟩
Main	2	⟨－,a002,Main⟩,⟨Main,a003,－⟩,⟨Main,a004,－⟩,⟨Main,a005,－⟩,⟨Main,a006,－⟩
PublishNews	3	⟨－,a007,PublishNews⟩,⟨PublishNews,a008,－⟩,⟨PublishNews,a009,－⟩
UserManager	4	⟨－,a010,UserManager⟩,⟨UserManager,a011,－⟩,⟨UserManager,a012,－⟩
NewsEditor	5	⟨NewsEditor,a013,－⟩
EditNews	6	⟨－,a014,EditNews⟩,⟨EditNews,a015,－⟩
ChangePassword	7	⟨ChangePassword,a016,－⟩
ChangeUserName	8	⟨ChangeUserName,a017,－⟩
……	……	……

4　测试用例的生成

一个测试用例可以认为是一个二元组 $t=(In,Out)$，它是能导致程序一次执行的输入集，其中，In 表示输入值，Out 表示期望输出。下面给出的测试用例忽略了测试的期望输出，只给出了测试的输入值，可以看作是抽象的测试用例。

在进行测试时，一个重要的问题是先给出测试准则。一个测试准则是评估一个测试用例集的有用性的一种方式，该测试用例集根据规格说明来测试程序，诊查存在的不满足测试需求的错误。这样，一个测试准则可以被形式化地定义为 $S\times R\times\mathbb{P}(T)$ 的子集，其中 S 是规格说明的集合，R 是需求的集合，T 是测试用例的集合，$\mathbb{P}(T)$ 表示 T 的幂集。令 c 代表任意一个测试准则，$c(s,r,t)$就表示 t 根据 s 测试 r 时满足测试准则 c，即 t 被评估为充分的，其中 $c\in C,s\in S,r\in R,t\subseteq T$。测试准则集 C 可以被看成是一个全函数 $S\times R\times\mathbb{P}(T)\to$ {true,false}，即 $S\times R\times\mathbb{P}(T)$ 中的每个元素的取值要么为 true，要么为 false。$c(s,r,t)$表达的意思是内涵的，它表示(s,r,t)是 C 中的一个元素。

测试准则一般分为基于程序的(白盒的，结构化的)和基于规格说明的(黑盒的，功能的)。基于程序的测试准则(即，考虑程序的内部结构)又分为数据流测试准则和控制流测试准则。本文的测试准则可以看成是基于构件的测试准则，它是基于规格说明的。

在 Web 应用程序中，构件要支持分布式访问，在测试时，一般要测试构件的单节点访问和多节点并行访问。本文提出的模型和测试用例生成只涉及到单节点情况。在 Web 应用程序中有些构件有依赖关系或约束关系，是按顺序操作的。在顺序执行构件时，组合顺序可能有很多种，例如对于构件 A，B 和 C，先执行 A，再依次执行 B 和 C，得到执行序列 A→B→C，用序偶⟨A,B,C⟩表示；或按 B→A→C、C→A→B 的顺序执行等，共 6 种情况。我们把抽象测试用例看成是由构件组成的执行序列。因此，得到下面两个重要概念。

定义 1　在 FSM 中，执行序列是由构件的先后执行次序组成的序列。

定义 2　在 FSM 中，完整执行序列是从初始状态到终止状态的执行序列。

根据定义 2，我们可以给出完整执行序列覆盖测试准则。

定义 3(完整执行序列覆盖)　对于一个 FSM 模型，一个测试集 TS 满足完整执行序列覆盖，当且仅当对于 FSM 的任意一条完整执行序列，$\exists t\in TS$ 经过它。

例 1　对于图 2 构件依赖图的顶层 FSM 模型，构件 Login 对应的状态为初始状态，Exit 对应的状态为终止状态。可以得到测试用例集 TS={⟨Login,Main,Exit⟩、⟨Login,Main,PublishNews,Exit⟩、⟨Login,Main,UserManager,Exit⟩}。显然 TS 满足完整执行序列覆盖。

结合表 2，执行测试集 TS 中的每个测试用例，实际上是先后执行测试用例中的每个构件的动作，在执行的过程中检测 Web 应用的错误。此外在例 1 中，由于构件 PublishNews 是复合构件，包括构件 NewsEditor 和 EditNews。根据表 1，它们的关系类型为 $T2$，且存在交互(如图 3 所示)，则测试用例⟨Login,Main,PublishNews,Exit⟩可以写成⟨Login,Main,NewsEditor,EditNews,Exit⟩，但不能是⟨Login,Main,EditNews,NewsEditor,Exit⟩，因为根据图 3，必须先执行 NewsEditor，然后执行 EditNews；构件 UserManager 也是复合构件，包括构件 ChangeUserName 和 ChangePassword。根据表 1，它们的关系类型也是 $T2$，但它们之间不存在交互(如图 4 所示)，则测试用例⟨Login,Main,UserManager,Exit⟩可以写成⟨Login,Main,ChangeUserName,Exit⟩或者⟨Login,Main,ChangePassword,Exit⟩。

因此，满足完整执行序列覆盖的测试用例集 TS 被细化为两个测试用例集。它们是：

$TS1$={⟨Login,Main,Exit⟩、⟨Login,Main,NewsEditor,

EditNews,Exit)、(Login,Main,ChangeUserName,Exit)}和

$TS2$={(Login,Main,Exit)、(Login,Main,NewsEditor,EditNews,Exit)、(Login,Main,ChangePassword,Exit)}。

测试用例集 $TS1$ 和 $TS2$ 都满足完整执行序列覆盖测试准则,但是这两个测试用例集中无论哪一个都不能使得所有的构件得到测试。例如,$TS1$ 不能测试构件 ChangePassword,而 $TS2$ 不能测试构件 ChangeUserName。因此,为了覆盖所有的完整执行序列,保证所有的构件至少被测试一次,我们又提出下面的测试准则。

定义 4(构件完整执行序列覆盖) 对于一个 FSM 模型,一个测试集 TS 满足构件完整执行序列覆盖,当且仅当对于 FSM 的任意一条完整执行序列,$\exists t \in TS$ 经过它;而且所有的构件也至少被测试一次。

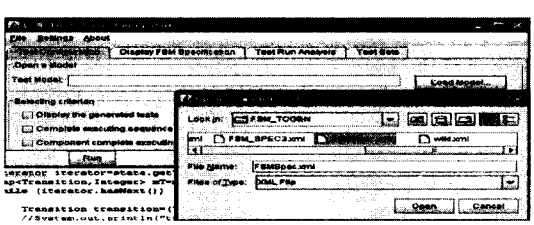
例 2 测试集 $TS12 = TS1 \cup TS2$ = {(Login,Main,Exit)、(Login,Main,NewsEditor,EditNews,Exit)、(Login,Main,ChangeUserName,Exit)、(Login,Main,ChangePassword,Exit)}就满足构件完整执行序列覆盖测试准则。

5 测试原型的实现

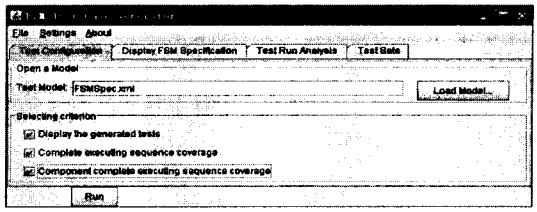
根据前面提出的方法,本文开发了一个测试原型 FSMTCGen(FSM Test Case Generator),生成满足一定测试准则的测试用例集。该工具使用 Java 语言在 Eclipse 环境下编写。我们将 Web 应用按功能进行划分,得到构件依赖图,并根据需求得到构件关系表,进而得到它的 FSM 测试模型,该测试模型使用 XML 文件存储。这样的 XML 文件实际上就是 FSM 测试规格说明。

测试原型 FSMTCGen 主要包含 MainWindow、LoadModel、ModelParse、ModelRun 和 ExportAbstractTestCase 这 5 个模块。其中,MainWindow 模块显示主界面,用户可以在界面上选取不同的覆盖测试准则。LoadModel 模块装载 XML 格式的模型文件(FSM 测试规格说明),并检查文件的合法性。两个核心模块是 ModelParse 和 ModelRun 模块。ModelParse 模块用于解析 XML 规格说明,用不同的数据结构装载模型,生成不同的中间过程对象,便于后期测试用例的生成。ModelRun 模块根据用户所选取的不同测试准则,利用自身封装好的执行引擎产生出一些中间执行序列,并通过各个监听器告知 ExportAbstractTestCase 模块,说明自己的执行序列已经构造成功。ExportAbstractTestCase 模块根据得到的消息,输出测试用例,例如打印到屏幕或者输出到文件。

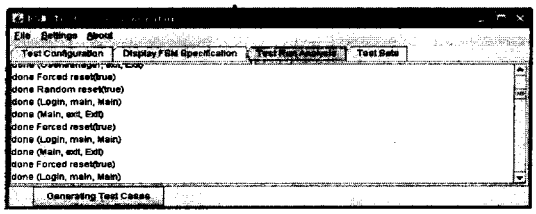
在 FSMTCGen 的主界面下,点击"Load Model…"按钮弹出装载模型文件对话框:



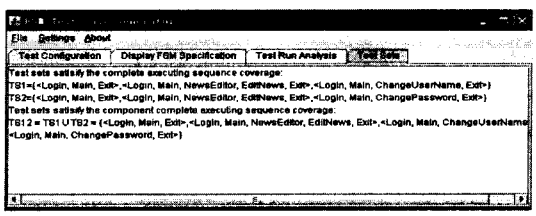
选定好模型文件后,接着需要指定测试准则,以便下一步测试执行的分析。



点击"Run"按钮可以查看得到的执行序列。



点击"Generating Test Cases"就得到了满足给定覆盖测试准则的测试用例集。



上面的过程首先装载模型文件,然后解析模型文件,接着产生出中间执行序列,最后给出满足一定测试准则的抽象测试用例集。通过该测试原型可以方便快捷地产生出测试用例集,也能帮助测试人员了解测试流程,简化了测试工作。

结束语 Web 应用的异构性、动态性、连接的多样性、控制流程的可变性以及需要快速开发与发布等特性给 Web 应用的建模和测试带来了新的挑战。本文按功能将 Web 应用划分成若干个功能构件,基于功能构件的概念对 Web 应用建模并进行测试,用 FSM 来表示功能构件的行为关系,用 FSM 的复合来表示构件的交互操作。根据提出的完整执行序列覆盖、构件完整执行序列覆盖这两个测试准则,生成相应的测试用例集。此外,还设计了一个测试用例生成的工具原型。然而,Web 应用本身的复杂性使得其建模和测试变得有些困难,本文的研究还存在一定的不足,下一步的工作包括 Web 浏览器提供的功能对构件交互测试的影响以及工具原型的进一步完善等。

参 考 文 献

[1] Stout G A. Testing a Website: Best Practices. A Whitepaper [OL]. http://www.reveregroup.com,2008-6-28

[2] Hieatt E, Mee R. Going Faster. Testing the Web Application [J]. IEEE Software,2002,19(2):60-65

[3] Brim L,Cerna I,Varekova P,et al. Component-Interaction Automata as a Verification-oriented Component-based System Specification[C]//Proc. of SAVCBS'05. Lisbon,Portugal,Sept. 2005:31-38

[4] Andrews A,Offutt J,Alexander R. Testing Web Applications by Modeling with FSMs [J]. Software Systems and Modeling, 2005,4(3)

(下转第 169 页)

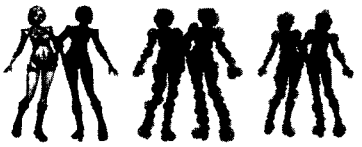


图 5　虚拟场景中不同层次包围球树的重建情况

4.3　实验 3：复杂碰撞情形的测试

在碰撞检测过程中，如何处理虚拟场景中物体的穿透问题，怎样完成穿透图像的重建，一直是研究的热点。本文提出的算法同样适合复杂的碰撞检测过程。图 6 给出了存在碰撞的穿透响应过程，第 2 和第 3 列分别给出了包围球树的第 5 层和第 7 层的重建过程。从图 6 可以看出，对于复杂的碰撞过程，本文提出的算法也是适用的。

图 6　虚拟场景中复杂情况的处理

结束语　本文提出了一个基于球体混合变形模型的碰撞检测算法，对于所有的变形模型，不需要考虑物体表面的变形情况便可以快速而精确地完成碰撞检测。实验证明，球体混合重建算法要求上限是很低的，这个算法健壮性及与其它算法的兼容性也很好。它可以通过球体混合重建和减小变形的方法在两个物体间发现碰撞。本文提出的关键技术是在单元四元组球上采用变换包围盒的方法实现碰撞检测，在需要重建的包围球上重建数据结构。算法的唯一限制是要求顶点数目的复杂性是线性的(对于相关顶点的数目，实际上它只要求子线性复杂性就可以了)。在未来的研究中，将考虑在线性复杂性的条件下，将这种混合技术应用在其它的几何模型中实现碰撞检测。

参 考 文 献

[1] Hoff K, Zaferakis A, Manocha D. Fast and simple 2D geometric proximity queries using graphics hardware[C]// Int'l Conf. Interactive 3D Graphics. Berlin ACM Press, 2001: 145-148

[2] 范昭炜，万华根，高曙明. 基于流的实时碰撞检测算法[J]. 软件学报，2004，15(10)：1505-1514

[3] Zhao Wei, He Yanshuang, Li Wenhui. Research on Parallel Collision Detection Algorithm Based on Pipelining and Divide-and-Conquer[C]// Int'l Conf. Intelligence Computation and Application. Wuhan: China University of Geosciences Press, 2007: 277-282

[4] 赵伟，何艳爽. 一种快速的基于并行的碰撞检测算法[J]. 吉林大学学报：工学版，2008，38(1)：152-157

[5] 魏迎梅，吴泉源，石教英. 碰撞检测中的固定方向凸包包围盒的研究. 软件学报，2001，12(7)：1056-1063

[6] Brown J, Sorkin S, Latombe J. Real-time simulation of deformable objects[J]. Computer Animation, 2001, 10(6): 228-236

[7] Redon S, Kim Y, Lin M C, et al. Interactive and continuous collision detection for avatars in virtual environments[C]// Proc. of the 3th IEEE Int'l Conf on Virtual Reality. Cambridge: IEEE Computer Society Press, 2004: 117-124

[8] Larrsson T, Moller T A. Efficient collision detection for models deformed by morphing[J]. The Visual Computer, 2003, 19(2/3): 164-174

[9] Klugt T, Alexa M. Bounding volumes for linearly interpolated shapes[C]// Proc. of the 22nd Int'l Conf on Computer Graphics. Crete, Greece: IEEE Computer Society Press, 2004: 134-139

[10] Mohr A, Gleicher M. Building efficient, accurate character skins from examples [J]. ACM Transaction on Graph (SIGGRAPH2003), 2003, 22(3): 562-568

[11] Heim O, Marshall C S, Lake A. Fast collision detection for 3D bones-based articulated characters [J]. Game Programming Gems 4, 2004(4): 503-514

[12] Hejl J. Hardware skinning with quaternions[J]. Game Programming Gems 4, 2004(4): 487-495

[13] Kavan L, Zara J. Fast collision detection for skeletally deformable models[J]. Computer Graphics Forum, 2005, 24(3): 363-372

[14] Eberly D. A practical approach to real-time computer graphics [J]. 3D Game Engine Design, 2001(9): 25-29

[15] Kavan L, Zara J. Spherical blend skinning: A real-time deformation of articulated models[C]// ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games. ACM Press, 2005: 9-16

[16] Gaertner B. Fast and robust smallest enclosing balls[C]// Proc. of the 7th Annual European Symposium on Algorithms. Prague: Springer Verlag Press, 1999: 325-338

[17] Brown J, Sorkin S, Bruyns C, et al. Real-time simulation of deformable objects: tools and application [C]// Proc. of the 14th Int'l conf on Computer Animation. Seoul, South Korea: IEEE Computer Society Press, 2001: 228-258

(上接第 127 页)

[5] Miao Huaikou, Chen Shengbo, Liu Huanzhou, et al. An Approach to Generating Test Cases for Testing Component-based Web Applications[C]// IITA 2007. IEEE Computer Society, 2007: 264-269

[6] 刘焕洲，缪淮扣. Web 应用程序建模和测试用例生成方法[J]. 计算机工程，2008，34(6)：60-62

[7] Kung D C, Liu Chien-hung, Hsia P. An Object-oriented Web Test Model for Testing Web Applications[C]// First Asia-Pacific Conference on Quality Software. New York, USA, Oct. 2000: 111-120

[8] Liu Chien-hung, Kung D C, Hsia P. Object-based Data Flow Testing of Web Applications[C]// First Asia-Pacific Conference on Quality Software. New York, USA, Oct. 2000: 30-31