

文章编号:1001-9081(2007)05-1277-03

## 农业专家系统的自动测试

尹来武

(吉林农业科技学院 信息科学系,吉林 吉林 132101)

(yinlaiwu@263.net)

**摘要:**针对中国农业专家系统研究中存在的问题,在消化和吸收国内外先进研究成果的基础上,结合现有的科研工作,对软件自动测试技术进行了一些研究和探索,并开发了一套农业专家系统自动测试工具。软件测试工具提供数据库测试、知识库测试、案例自动生成与维护、健壮性测试;可针对目前在农业专家系统开发中所广泛使用的面向对象编程技术、软构件技术所开发的一系列功能模块(主要是 .EXE 和 .DLL 文件)进行自动化的测试。

**关键词:**专家系统;自动测试;程序集合

**中图分类号:** TP182 **文献标识码:** A

## Auto-test for agricultural expert system

YIN Lai-wu

(Department of Information Science, Jilin Agriculture Science and Technology College, Jilin Jilin 132101, China)

**Abstract:** To deal with the problems existing in the research of the Agricultural Expert System, on the basis of digesting and absorbing the advanced research results home and abroad, we made some research and exploration at auto-test technology of software and developed an auto-test tool of Agricultural Expert System with reference to the present research work. This software test tool provides database test, knowledge base test, auto production of cases and maintenance and robustness test. It may do auto-test for a series of function modules (mainly include .EXE and .DLL files) developed by Object Oriented Programming and Soft Component Technology used extensively.

**Key words:** experts system; auto testing; program collection

### 0 引言

我国农业专家系统的研究始于 20 世纪 80 年代初期。随后,许多科研院所开展了各种农业专家系统的研究、开发及推广应用活动。现在已形成 5 个农业专家系统开发平台,智能应用示范区扩大到 23 个,各地开发的本地化农业专家系统近 200 个。目前专家系统的研究正在向广度和深度推进。农业专家系统软件的质量则成为开发者与使用者共同关注的焦点问题。

在农业专家系统软件开发过程中存在不少问题:开发手段、开发方式多种多样,导致软件复杂度急剧上升;开发人员、开发部门的软件开发技术水平参差不齐;测试手段、测试方法严重滞后。进行手动对农业专家系统软件进行测试将极大的降低软件测试的效率,而且测试的质量也难以保证。采用自动测试成为一种必然。

为了更好地测试农业专家系统软件,进一步提高农业专家系统软件测试床的测试效率,本文研制开发了一个软件自动测试工具,用来衡量农业专家系统软件的质量优劣。

### 1 农业专家系统软件自动化测试

我们主要从软件设计和程序编码的角度对农业专家系统软件进行系统分析,从而确立自动测试的目标,提出相应的自动测试策略,建立农业专家系统软件自动测试的框架。

#### 1.1 关于程序集合

农业专家系统软件的开发手段与开发方式多种多样,数据库技术、网络技术、智能模拟技术等都融入了农业专家系统

的开发过程中。同时,VC、VB、C#和 Java 等一大批开发语言正逐渐成为农业专家系统软件开发的主流编程语言。在开发农业专家系统软件的过程中,农业专家系统的通用性、可移植性越来越受到人们的关注;面向对象技术,软构件技术正逐渐成为提高软件开发效率的重要开发手段。

由于农业专家系统软件的复杂程度越来越高,为了缩短开发时间,提高开发效率,软件开发人员经常以模块化的方式进行程序的编码。

采用功能模块这种编程方式进行程序设计与开发,主要是因为程序编码的工作量巨大,必须由多个软件开发人员根据软件设计规范分头实施,这种将一个大的应用程序划分为几个小模块的构建方式,有利于软件开发小组内部成员的分工与合作。

这些功能模块从软件编程的角度看,又可以分为两类:一类直接嵌套在农业专家系统软件中;另一类具有独立性,即:经过软件的代码编写,并对它们进行软件的编译之后,以 \*.exe 或 \*.dll 为后缀的文件形式存在,一般被称为程序集合。

本文主要针对程序集合进行自动测试。下面主要讲述程序集合的特点,以及其在农业专家系统中的应用情况。

图 1 简单表达了程序调用过程。在程序编码过程中,这种编程方式是大量存在的。在系统运行过程中,用户的输入数据或系统事件产生的输入数据,通过程序集合中提供的接口传入,在程序集合中进行相关的处理,并输出结果。

从农业专家系统软件编程的角度看,这种程序集合的编程方式被广泛地运用在推理机、知识库、模型库的开发过程

中,而推理机、知识库、模型库是农业专家系统软件不可或缺的组成部分,发挥着重要的作用。

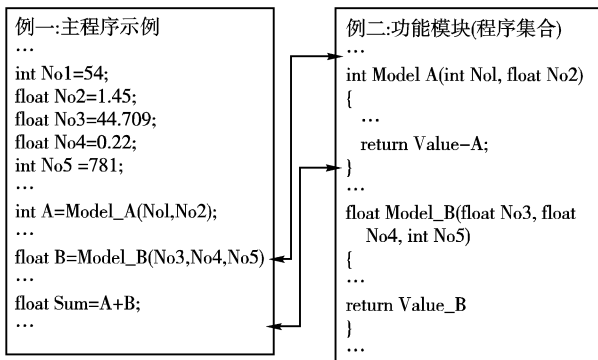


图1 程序调用过程

从推理机、知识库、模型库运行机理进行分析,可以看出:农业专家系统软件一般都包含有推理机、知识库、模型库,它们包含了大量的农业领域的相关知识;并且通过软件编程技术手段,把农业相关领域的知识按照一种合理的表示方法将它们移植到计算机中。当使用者需要使用这些知识处理问题时,一般以发送信息的形式调用相应的功能模块进行运算,并得到一个对应的运算结果。如图2所示。

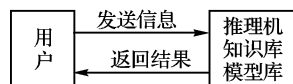


图2 程序调用机理

可以看出,返回结果的正确与否,将直接关系后面继续运行的应用程序的成败,进而导致整个农业专家系统软件运行的成败。返回结果是由推理机、知识库、模型库中相应的数据和算法的逻辑运算实现的。因此,必须对推理机、知识库、模型库进行大量的测试,即:通过发送信息,观察返回结果来发现它们存在的数据和算法的逻辑运算错误。

从软件编程的角度对推理机、知识库、模型库进行分析,可以发现:推理机、知识库、模型库中都包含有大量的处理实际功能的程序集合,大多数的程序集合的开发都遵循面向对象的软件编程技术规范。即:每一个程序集合都包含一个或多个类,每一个类中又包含多个方法,而且具体功能的实现又主要通过这些方法的调用运行去完成。程序集合中信息与数据的交互通过接口(在程序编码中表示为公共属性 <public> 的方法函数)进行传递。

例一:

```

...
类1
{
...
方法1(变量1,变量2,...变量n)
{
...
return 返回值
}
...
方法n(变量1,变量2,...变量n)
{
...
return 返回值
}
...
}
...

```

```

类 n
{
...
方法1(变量1,变量2,...变量n)
{
...
return 返回值
}
...
方法n(变量1,变量2,...变量n)
{
return 返回值
}
...
}

```

上例简单的展示了程序集合内部实际的程序编码方式。可以看出:一个具体功能的实现,主要通过方法的调用来完成,每一个方法的内部由大量的程序代码组成,通过程序代码的运行完成一个具体功能的运算操作过程。其入口是方法,方法以接口的形式与外界进行联系,这些接口由一个或多个不同类型的参数变量组成,提供给用户向其发送信息,从而进行相关的运算;出口是方法的返回值,用来向调用者返回结果。因此,在这样的层次中,程序集合所包含的类中所有的方法都是需要进行测试的对象单元,即:通过对方法的测试,查找其编程中产生的逻辑算法是否存在错误。

综上所述,为了保证整个农业专家系统软件的安全性、健壮性和可用性,就必须对程序集合进行足量的测试,以保证软件的质量。

## 1.2 程序集合自动测试

农业专家系统软件中程序集合的编程特点决定了测试这种类型软件的难度大,工作量大,需要测试的对象多,测试工作复杂繁重。而对程序集合这种软件系统进行自动测试将有助于达到其提高测试质量,增进测试效率的要求。因为:

- 1) 自动测试具有一致性和可重复性;
- 2) 自动测试可以执行一些手工测试困难或者不可能做的测试;
- 3) 自动测试可以发挥计算机的巨大优势。

软件的自动测试就是要做到在最大限度的减少软件测试人员的负担下,提高测试的效率,提高测试的质量。

## 1.3 自动测试策略

对程序集合的自动化测试,概括的讲就是根据程序集合接口的特点,向其输入相应的测试数据,并自动运行,对其运行完毕后的返回值进行比较、分析,以发现程序集合中的问题,发现错误、缺陷,便于进行修改。

农业专家系统软件的自动化测试,其最终是要通过建立自动化的测试工具来实现。在研制、开发自动化测试工具的过程中,一定要针对测试的内容,对测试框架进行细致、周密的考虑,使尽可能多的测试过程通过自动测试工具来完成,尽量减少手工测试的步骤、过程,即:实现自动化测试的最大化。

### 1) 自动获取被测对象信息

通过对农业专家系统软件的分析,可以发现:农业专家系统软件中包含多个程序集合,每个程序集合中又包含有大量的需要被测试的方法,每一个被测试的方法又由一到多个变量组成数据输入接口;因此,自动测试工具如果能自动捕获这些“关键”测试信息,并以此为依据产生测试案例,将会提高软件的测试效率。这是本论文所要解决的一个关键技术问题。

### 2) 自动生成测试案例

根据捕获的被测试对象的信息,根据软件测试人员的测试需求,自动产生完整、精确的测试案例。测试案例自动存储在测试工具的案例库中,为被测试程序经过修改后的重新测试提供了大量的数据资料。

### 3) 自动创建测试脚本

如果为每一个被测试对象、每一个测试案例都编程一个测试脚本,这将大大降低测试的效率,也违背了自动化测试的初衷。因此,根据被测试对象的特点,自动生成测试脚本,而无需软件开发人员手工编程,将会提高软件的测试效率。这是本论文所要解决的另一个关键技术问题。

自动创建的测试脚本应包括以下主要功能:自动连接被测程序集合、根据已经捕获的被测试对象信息自动产生驱动程序编码、自动启动被测试程序集合、在测试过程中,跟踪、监视被测试程序的运行,收集产生的错误信息和实际的返回信息。

### 4) 自动测试策略流程

实现自动测试策略的流程如下:

- 自动获取被测程序集合中的信息;
- 自动产生测试案例;
- 自动生成测试脚本;
- 以测试脚本驱动被测程序集合,进行测试,并进行结果的自动比较与分析。

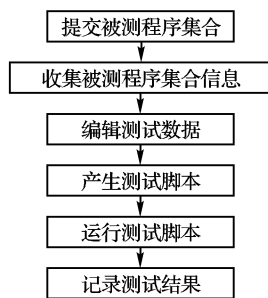


图 3 自动测试策略

(1) 被测程序集合提交给自动测试工具之后,自动测试工具具有自动获取被测对象的功能,即:自动测试工具可以自动搜索被测程序集合中所有的方法,并把搜索到的方法的方法名,方法中包含的变量名,变量类型清晰的显示出来,便于下一步的操作和运行;

(2) 测试人员选择一个方法作为被测试对象之后,自动测试工具可针对所选择的对象特点,自动产生测试人员所需要的测试案例;

(3) 之后,自动测试工具根据产生的测试案例,自动创建测试脚本并自动执行测试脚本,对被测对象进行自动化测试;

(4) 在自动测试过程中,测试脚本一方面把测试数据传入被测方法中(通过接口调用的方式);另一方面,测试脚本时刻监视被测程序集合的运行,并把测试结果输出,显示测试结果,便于查找缺陷;

(5) 自动测试工具可自动存储使用过的测试案例、测试脚本;当进行回归测试时,可直接使用这些以前已经使用、生成的部件,从而提高了测试的效率。

由于需求、规格和代码的不断改变,增加、删除或修改代码都需要测试,代码的改变和进化通常是持续的,有时开发者并不了解发生的所有改变,因此,只有进行测试才能确认代码的改变是否有效,新的或者修改的功能只有通过测试验证后,才能集成到系统中,从而确保这些改变不会导致系统崩溃。例如:对公有类接口和组件接口进行测试,以确保修改后的编

码不影响接口协议。

当被测成员不太多时,进行手工测试,手工编码测试脚本还比较容易,但对于有许多类和成员组成的程序集合,手工编码所有的测试脚本将会耗费大量时间,而且,为了精确编码测试脚本,需要研究被测程序集合的每个方法。

测试工具的测试过程不需要人工干预,而是连续执行,并能把揭示的缺陷及时通知测试人员。这对于自动测试过程是非常有用的,而且使回归测试也更易于管理。一方面,测试工具基于存储数据完成单元验证测试;另一方面,基于预定方案,可以重新运行产生的测试脚本,实现回归测试,从而判断修改后的应用程序能否满足需求。

## 2 系统运行实例

### 2.1 新的测试

“新的测试”主要完成以下功能:

- (1) 自动获取被测试对象信息(包括:被测试的程序文件名、类名、方法名、变量名);
- (2) 确定每一个变量的取值范围;
- (3) 确定需要生成的测试案例的数值;
- (4) 确定测试期望值的范围。

### 2.2 测试案例显示

本界面显示“即将”生成的测试案例所包含的内容。软件测试人员可以修改、添加、删除测试数据,从而使测试案例满足测试的需要(如图 4 所示)。



图 4 测试案例浏览界面

### 2.3 结果显示

自动测试完毕后,自动测试工具把测试的结果显示出来,便于测试人员进行分析、处理。

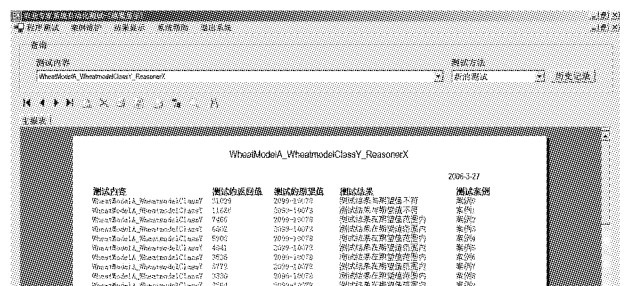


图 5 测试结果显示界面

## 3 结语

针对农业专家系统软件的特点,提出针对采用面向对象技术开发的程序集合进行测试的测试方法与解决对策;并以自动获取测试对象信息、自动生成测试脚本为本论文的研究重点,在 C#. Net 开发环境下实现了软件自动测试工具的开发。

通过大量的实际程序测试的实践证明,按照本文提出的软件测试方法使得对农业专家系统软件软件测试工作由盲目变为有序,测试目的性强,测试效率高,有效地缩短了软件 (下转第 1282 页)

$$F(t) = \{k \times [1 - e^{-(\frac{t}{a})^2}]\} \quad (1)$$

其中,参数  $a$  表示 NPC 的生命力值;参数  $k$  表示 NPC 的类型,不同的 NPC 对同一游戏环境的适应性是不一样的,当  $k$  取不同的值时,会得到适应度不同的各种 NPC。接着按照以下工作步骤操作:

- 1) 从 NPC 神经网络中提取权重向量;
- 2) 用遗传算法演化出一个新的网络权重群体;
- 3) 把新的权重插入到 NPC 神经网络;
- 4) 转到第一步进行重复,直至获得理想的性能。

#### 4 试验分析

我们的实验测试场景如下:

在一个仿真的三维游戏环境下,游弋着若干个 NPC 角色和一个玩家控制的虚拟角色,主角可以漫游整个游戏场景,这些 NPC 当遇到主角后,可能会对主角采取不同的行为,比如攻击行为,逃避行为,团队作战行为,对话行为等,所有这些行为的决策都取自于神经网络的训练。

在采用神经网络算法之前,所有的 NPC 无论强弱,都会主动向玩家角色发起攻击,而在采用神经网络算法之后,这些 NPC 都具有了一个人工大脑,每个 NPC 在与玩家角色的交互过程不断地学习,不断地演化,最终变成自主角色,具体表现在: NPC 根据以往与玩家角色交互过程中的经验,从而产生较为理智的行为,比如当 NPC 感觉玩家的综合实力要高于自己时,它可能会采取逃避的行为,而当 NPC 感觉其综合实力要高于玩家时,它往往会主动攻击玩家。

表 1 和表 2 列举了应用神经网络算法前后的测试数据。

表 1 应用神经网络算法之前的测试数据(部分)

输入							输出动作	NPC 最终状态
NPC 攻击力	NPC 生命力	NPC 防御力	NPC 情感值	玩家 攻击力	玩家 生命力	玩家 防御力		
0.8	0.8	0.6	0.5	0.8	0.4	0.3	攻击	存活
0.3	0.5	0.4	0.4	0.9	0.70	0.6	攻击	死亡
0.5	0.4	0.5	0.1	0.5	0.6	0.4	攻击	死亡
...	...	...	...	...	...	...	...	...

应用神经网络算法所采取的实验方案如下:

(1) 对于 NPC 感知系统的输入,包括与虚拟玩家角色的距离,虚拟玩家的攻击力,防御力,生命力,魔法力,信誉度, NPC 自身的攻击力,防御力,生命力,魔法力,信誉度。并将参数归一化,使最终的参数范围位于  $[-1, 1]$  之间;

(2) 对于 NPC 感知系统的输出,包括躲避,单独攻击,潜伏,召唤同伴,团队攻击等行为。即将神经元的输出元设计成

五个,这些行为通过一些标志符来确定,例如,如果代表攻击的输出位为 1,则其他位为零。

通过对比两组测试试验,可以发现后一组试验中, NPC 能够根据自己的实力和玩家的实力对比,理智的采取一些行为(比如退避,呼唤同伴协同作战)而不是一味盲目攻击, NPC 的存活率显然就很高,因此也显得较为智能。

表 2 应用神经网络算法之后的测试数据(部分)

输入							输出动作	NPC 最终状态
NPC 攻击力	NPC 生命力	NPC 防御力	NPC 情感值	玩家 攻击力	玩家 生命力	玩家 防御力		
0.8	0.8	0.6	0.5	0.8	0.4	0.3	攻击	存活
0.3	0.5	0.4	0.4	0.9	0.70	0.6	躲避	存活
0.5	0.4	0.5	0.1	0.5	0.6	0.4	漠然置之	存活
...	...	...	...	...	...	...	...	...

#### 5 结语

本论文通过认知建模研究了 NPC 的自主性,采用神经网络和遗传算法相结合的游戏自主角色的设计思路,利用遗传算法优化神经网络的方法设计了一个自主角色的框架,建立了一个游戏角色的自学习模型,分别对比应用神经网络算法之前和之后的两组仿真实验,结果表明采用神经网络和遗传算法相结合的角色生存能力要明显好于基于传统算法形成的角色的生存能力,这进一步说明采用神经网络和遗传算法相结合的非确定性算法生成的游戏角色自学习系统可以使 NPC 成为更加自主和智能的角色。

#### 参考文献:

- [1] REYNOLDS CW. Flocks, Herds, and Schools: A Distributed Behavioral Model[J]. Computer Graphics, 1987, 21(4): 25-34.
- [2] 方约翰. 人工智能在计算机游戏和动画中的应用—认知建模方法[M]. 北京: 清华大学出版社, 2004.
- [3] TU X, TERWPOULOS D. Artificial fishes: Physics, locomotion, perception, behavior[A]. Proceedings of SIGGRAPH94[C]. Orlando, Florida, 1994.
- [4] LUMBERG BMB, ALYEAN TAG. Multi-level direction of autonomous creatures for real-time virtual environments[A]. Proceedings of SIGGRAPH 1995[C]. 1995. Los Angeles, California, 1995.
- [5] MCCARTHY J, HAYES P. Some philosophical problems from the standpoint of artificial intelligence[A]. MEHZER B, MICHIE D. Machine intelligence[C]. Edinburgh, Scotland: Edinburgh University Press, 1969. 463-502.
- [6] 陈国良, 王煦法, 庄镇泉, 等. 遗传算法及其应用[M]. 北京: 人民邮电出版社, 1999.
- [7] [J]. 计算机研究与发展, 2001, 38(6).
- [4] 刘金艳. 面向对象软件测试技术的研究[D]. 成都: 西南交通大学, 2003.
- [5] 朱少民. 软件测试方法和技术[M]. 北京: 清华大学出版社, 2005.
- [6] 朱小骏, 高建华. 一种面向对象程序系统的有效测试方法[J]. 计算机工程与设计, 2004, 25(7).
- [7] KANER C, BACH J, PETTICHORD B. 软件测试经验与教训[M]. 韩柯, 译. 北京: 机械工业出版社, 2003.
- [8] WHITTAL JA. 实用软件测试指南[M]. 马良荔, 俞立军, 译. 北京: 电子工业出版社, 2003.
- [9] [http://www.51testing.com/\[EB/OL\]](http://www.51testing.com/[EB/OL]), 2006-09-09.
- [1] 黄陇, 陈致明, 于洪敏, 等. 基于 UML 的软件测试用例自动生成技术研究[J]. 计算机应用与软件, 2004, 21(11).
- [2] 姜文, 缪准扣, 刘玲. 一个自动构造类测试驱动程序的框架[J]. 计算机科学, 2005, 32(3): 159-161, 185.
- [3] 李留英, 王戟, 齐治昌. UML Statecharts 的测试用例生成方法

(上接第 1279 页)

项目的开发周期。

另外,在实际测试工作中,不应该只“单一”的采用手工测试或自动测试,而应该将它们有机结合起来,取长补短,充分发挥各自的特点。

#### 参考文献: