

文章编号:1001-9081(2010)02-0374-06

# 场景—事件驱动的安全苛求软件系统仿真测试脚本语言研究

喻 钢<sup>1,2</sup>, 徐中伟<sup>1</sup>, 杜军威<sup>1,3</sup>

(1. 同济大学 电子与信息工程学院, 上海 201804; 2. 上海大学 悉尼工商学院, 上海 201800;

3. 青岛科技大学 信息科学技术学院, 山东 青岛 266061)

(yugang509@163.com)

**摘 要:**脚本技术是实现软件测试自动化技术的有效方法。软件测试脚本化可以降低测试成本,提高测试效率。测试脚本语言是脚本技术的核心,但是目前没有专门针对安全苛求软件测试的脚本语言。因此结合安全苛求软件测试的特点,提出了场景—事件驱动的针对安全苛求软件仿真测试的通用测试脚本语言 SED\_SCS\_STL,对该语言的设计以及在测试环境中的实现机制进行了阐述,并将其应用于高速铁路 CTCS-2 级车站列控中心的软件仿真测试中。

**关键词:**安全苛求软件;测试脚本语言;安全性测试

**中图分类号:** TP311.53 **文献标志码:** A

## Research on scenario-event-driven simulation test script language for safety-critical software system

YU Gang<sup>1,2</sup>, XU Zhong-wei<sup>1</sup>, DU Jun-wei<sup>1,3</sup>

(1. College of Electronics and Information Engineering, Tongji University, Shanghai 201804, China;

2. Sydney Institute of Language and Commerce, Shanghai University, Shanghai 201800, China;

3. School of Information Science and Technology, Qingdao University of Science and Technology, Qingdao Shandong 266061, China)

**Abstract:** Script technique is an effective method to implement the automation of software testing. Using test script reduces the cost for the software test and improves the efficiency of software test. Test script language is the core of script technique, but the special test script language for safety-critical software test does not exist. Therefore based on the feature of the safety-critical software testing, the concept of safety-critical software simulation test script language (SED\_SCS\_STL) driven by scenario-event was presented in this paper. The design of the language and how the language was implemented on the safety-critical software simulation test platform was elaborated in detail. Finally the application of the language in the simulation and test of the train control center of CTCS-2 high speed railway control system was shown.

**Key words:** safety-critical software; test script language; safety test

## 0 引言

在诸如高速铁路、航空航天、核工业等应用领域的系统通常被称为安全苛求系统(Safety-Critical System)<sup>[1]</sup>,因为它一旦失效,将引发重大财产损失或者人员伤亡。而作为安全苛求系统的核心,其软件系统亦被称为安全苛求软件系统(Safety-Critical Software System)。对于这类安全苛求软件系统的安全性测试是验证系统功能和确认系统具有足够安全防护能力的重要手段,测试结果也是对此类系统进行安全性评价的重要依据。由于安全苛求软件系统中安全性测试的量非常大,测试案例需要涵盖的内容相当广泛,人工测试的费用相当可观,唯有采用能降低人工开销的自动测试才能使安全性测试成本控制在合理的程度,提高测试效率。

脚本技术是实现软件测试自动化的有效手段<sup>[2]</sup>。测试脚本语言是脚本技术的核心,采用测试脚本语言对测试用例给以形式化的描述,通过构建可自动执行的脚本,可以提高软件测试的质量和可维护性,大大减少测试人员的工作量,同时测试脚本的可重用性可以提高测试的可重复性。

## 1 研究问题分析

在国内外,测试脚本语言领域的研究正在不断发展,并在

实际中得到应用。在软件自动化测试中主要使用以下几种脚本技术<sup>[3]</sup>:线性脚本、共享脚本、数据驱动脚本、关键字测试脚本。比较典型的引入脚本技术的测试工具有英国 IPL 公司 Cantata ++ 测试工具集、美国 Rational 公司 Visual Test 测试工具、Mercury 公司的 WinRunner 和 LoadRunner、以色列 Technion 大学的 T++ 测试工具、美国喷气推进实验室(JPL)开发的针对空间飞行器仿真测试的分布式集成环境和德国 Tech S. A. T 公司针对航天系统的开发环境 ADS2 等。在这些测试工具中,脚本完成测试功能的描述、测试流程的控制等功能,大大提高了软件测试的自动化程度。

虽然目前上述测试工具引入了脚本技术,但专门针对安全苛求软件系统的测试脚本语言尚不多见,而且没有引入针对系统安全特性和对各种故障的安全防护反馈处理机制的测试描述与测试控制机制。因为安全苛求软件具有安全性要求且需要对故障进行安全防护处理,这样就不能保证测试的闭环特性,不能真正地模拟被测软件系统的真实运行环境。

本文针对安全苛求软件系统的特点,提出了针对安全苛求软件系统仿真测试的测试脚本语言(Scenario-Event Driven Safety-Critical Software Simulation Test Script Language, SED\_SCS\_STL),对该语言的核心进行了详细阐述,并将其应用于高速铁路 CTCS-2 级车站列控中心软件系统的仿真测试中。

收稿日期:2009-08-01;修回日期:2009-09-14。

基金项目:国家自然科学基金资助项目(60674004);铁道部科技研究开发计划项目(2008X003-A;2009X002-A)。

作者简介:喻钢(1977-),男,江西南昌人,博士研究生,主要研究方向:安全软件形式化、建模、测试和仿真;徐中伟(1964-),男,江苏无锡人,教授,博士生导师,博士,主要研究方向:安全软件形式化、建模、测试和仿真。

## 2 SED\_SCS\_STL 的设计

目前,典型的安全苛求软件系统通常结构复杂,针对复杂多变的各种运营场景,它能随着运营场景的变换,根据外部环境发生的各种事件及系统内部状态,及时响应并以足够快的速度完成相应处理,并能针对各种故障依据相应的安全防护策略采用相应的防护操作。针对安全苛求软件的特点,对它的功能性与安全性测试一般采用黑盒测试及仿真测试的方法。

### 2.1 核心设计

测试脚本是测试过程中指导整个测试正常运行的“剧本”,是具有正规语法的数据和指令的集合。在软件测试过程中,测试脚本通常以文本形式存在,由测试脚本组织用户所施加的一系列软件执行动作,即一个测试脚本可以实现一个或多个测试用例的操作,达到驱动被测软件运行的目的,从而完成测试。和传统的脚本技术不同,SED\_SCS\_STL 定义的测试脚本以被测软件的实际运营场景为基础,将运营场景定义为相应的测试场景,在测试场景中以测试序列的模式,对相关测试用例进行组合,通过定义仿真测试环境的各仿真对象及仿真事件的抽象驱动,驱动仿真测试环境中的各仿真对象,将测试用例中包含的测试数据作为测试激励加载到被测系统中,完成各种功能性测试;同时也可以通过定义各种故障对象、安全防护机制及测试用例等,构建故障场景,完成各种安全性测试。图1为 SED\_SCS\_STL 的核心语法库。在核心语法库中,SED\_SCS\_STL 分为最底层的基础语言包、中间层的测试描述包及最外层的用户开发包。基础语言包主要提供 SED\_SCS\_STL 的各种基础语句及内置封装的库函数,为用户定义测试描述包中的各测试对象服务。测试描述包主要定义在 SED\_SCS\_STL 测试脚本中的各对象,SED\_SCS\_STL 将测试描述语言包中各部分封装成为对象或模板的形式提供给用户使用。用户开发包允许用户将实际仿真测试中使用到的各种测试脚本对象定义封装集成,以实现测试脚本的重用。



图1 SED\_SCS\_STL 核心语法库

基于 SED\_SCS\_STL 的上述语法规范,一个标准的 SED\_SCS\_STL 测试脚本结构如图2所示。采用 SED\_SCS\_STL 编写的测试脚本主要分为两层:测试主控单元模块脚本和测试执行单元模块脚本。在测试主控单元模块脚本中主要定义当前测试控制逻辑和当前仿真测试环境配置。测试执行单元模块脚本中主要定义当前测试中具体的执行测试内容,它分为场景级、测试序列级、测试用例级和仿真事件级等四层执行单元模块。各层次执行单元模块相互之间构成映射关系。场景级执行单元模块中主要定义当前测试场景配置及场景中需要执行的测试序列及执行逻辑。测试序列级执行单元模块主要定义当前测试序列中所需执行的测试用例及执行逻辑。测试用例级执行单元模块主要定义当前测试用例中所涵盖的测试

激励数据、预期结果及测试行为,在测试行为部分定义当前测试用例需要触发的相关仿真对象及触发执行逻辑。仿真事件级执行单元模块中主要定义当前仿真对象的仿真事件、相应的仿真对象的驱动命令及消息序列。

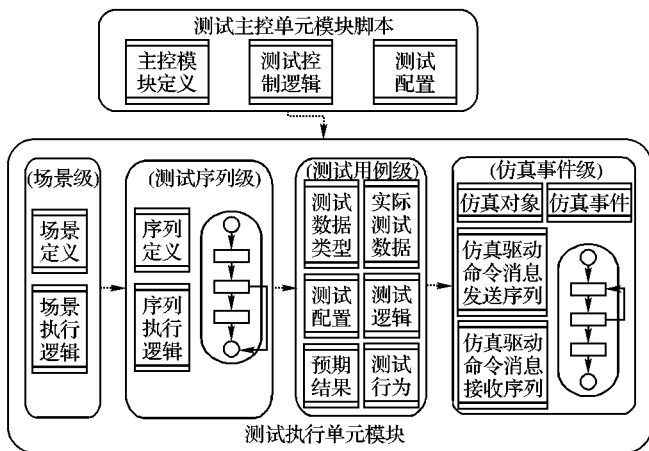


图2 基于 SED\_SCS\_STL 的测试脚本架构

传统的基于数据驱动测试脚本或关键字驱动测试脚本的自动化测试技术,通过数据和流程控制分离的脚本,读入数据文件来驱动流程进行,或脚本、数据、业务分离,数据和关键字在不同的数据表中,通过关键字来驱动测试业务逻辑进行系统功能测试,但是安全苛求软件系统的实际运营场景复杂,利用传统脚本技术进行测试,对于实际运营场景的考虑不完备,没有提供有效的描述机制,仅仅是完成单纯的功能性测试,未考虑在所有实际的运营场景中,尤其是复杂的运营场景下系统功能交互的正确性及故障场景下系统安全防护功能的正确性,而且测试逻辑并不是由运营场景变换来驱动,测试用例往往并不是通过仿真环境中各仿真对象的仿真事件驱动及仿真消息加载。因此,针对上述局限性,区别于传统的测试脚本语言,SED\_SCS\_STL 最显著的特点是“场景-事件驱动”,它的核心思想是以系统各种实际运营场景作为测试的核心,通过对于运营场景,包含正常运营场景与故障运营场景的描述,有机地将系统的功能特征点整合,进行测试。

“场景-事件驱动”体现在:

1) 核心语法设计。核心语法语言库中的测试场景对象专门用于描述系统在实际运营中可能出现的各种场景,而且测试场景对象具有可复用性,分为原子测试场景与复合测试场景。通过测试场景对象的描述来实现对某实际运营场景下系统的相关功能特征点及安全防护特性的测试。另一方面,在实际运营场景中,系统的功能往往是由与外界环境的不同交互事件的触发、不同消息的交互来体现,系统的安全防护性根据运营场景下外界环境的状态来触发。核心语法语言库中的仿真对象、仿真事件、仿真消息等对象提供了完整的系统外界环境描述机制用于描述系统的外部仿真对象、仿真事件及消息和测试用例加载机制。

2) 测试脚本架构设计。在测试脚本架构中,总体测试逻辑的驱动由测试场景对象建立的与实际运营场景——对应的测试场景级脚本的变换来实现,测试场景的变换真实地模拟了系统实际运营中可能出现的所有情况。在某个测试场景下,测试逻辑与测试用例的驱动由该测试场景下的各级低层测试脚本中定义的各仿真对象序列、仿真对象的事件序列、测试用例的加载通过仿真消息对象序列完成。

即形成了由测试场景驱动测试,由事件触发加载测试用

例,依靠此机制完成系统测试。

## 2.2 仿真测试执行机制

基于 SED\_SCS\_STL 的安全苛求软件测试过程一般可分为两个阶段:首先利用 SED\_SCS\_STL 语言描述可执行测试脚本(Executive Test Script, ETS),然后在仿真测试平台上运行 ETS,仿真测试平台中的脚本解释器根据测试脚本结构,将各级脚本之间及脚本元素之间的引用关系,将其中所有的脚本元素都转换成测试脚本语言对象执行树(Test Script Language Object Execute Tree, TSLOET)。其中根节点表示 SED\_SCS\_STL 测试主控模块的控制部分;最底层叶节点表示仿真事件级执行单元中的不可再分的原子操作,即向外部仿真环境中各仿真对象交互的各仿真驱动命令;中间节点则用于表示 SED\_SCS\_STL 其他各级执行单元脚本中的语句。脚本执行器根据 TSLOET,驱动仿真环境中的各仿真对象与被测系统(System Under Test, SUT)通信,并接收被测系统反馈,生成测试判决。图3是一个测试脚本语言对象树的表示。

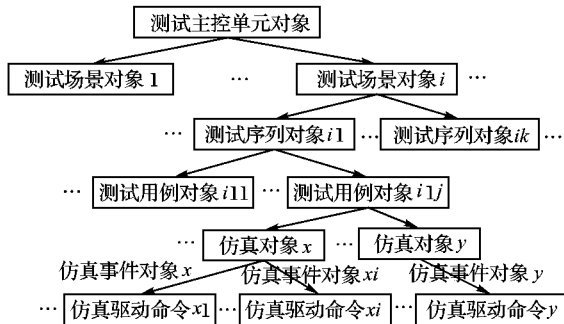


图3 测试脚本语言对象执行树

基于 SED\_SCS\_STL 的仿真测试脚本执行,测试脚本转换为对象执行树时,具体测试执行关键算法流程如下:

输入:对象执行树;

输出:仿真对象驱动命令(发送/接收消息);

处理:

- 1)对象树搜索,生成待处理测试场景队列;
- 2)如果测试场景队列为空,算法结束,否则转3);
- 3)获得测试场景队列首元,如果首元为空,算法结束,否则转4);
- 4)搜索对象树,建立当前测试场景下的测试序列队列,如队列为空,则转3),否则转5);
- 5)获得测试序列队列首元,如果首元为空,算法结束,否则转4);
- 6)搜索对象树,获得当前测试序列下的所有测试用例,建立测试用例缓冲池;如缓冲池为空,则转5),否则转7);
- 7)遍历测试用例缓冲池,逐一获得测试用例涉及的仿真对象序列,仿真事件序列,建立仿真对象的仿真事件触发器,并初始化;
- 8)遍历仿真对象序列,加载测试数据,生成驱动命令;
- 9)启动仿真事件触发器,发送驱动命令消息或等待反馈消息。

## 3 SED\_SCS\_STL 通用自动化仿真测试环境

基于 SED\_SCS\_STL 上述核心语法、脚本结构及执行机制,可构建如图4的通用仿真测试环境。整个通用仿真测试环境架构主要包括如下几部分:测试控制中心、测试引擎、仿真环境接口适配器集、被测系统外部仿真测试环境。被测系统外部仿真环境主要承担仿真被测系统的实际运营环境,所有的仿真对象以插件方

式集成构成外部仿真测试环境。测试控制中心主要完成测试用例生成、测试序列生成、测试场景生成、可执行测试脚本生成、测试进程实时监控及测试结果智能分析。测试引擎主要完成测试脚本解析、执行、控制、仿真驱动命令生成、注入、测试结果采集、判定与存储及测试执行流程管理控制。仿真环境接口适配器集主要用于处理测试引擎与 SUT 的外部仿真环境之间的消息通信和过程调用。测试引擎不与实际的 SUT 直接通信,而是将测试脚本中相应测试用例中所包含的测试激励数据以预定义的消息报文交给仿真环境接口适配器集,它会根据消息报文中所存储的仿真对象标记,自动选择相应的仿真对象插件接口适配器,再将消息报文按被测系统外部仿真环境中可识别的格式封装后通过网络发送给中相应的仿真对象插件,由仿真对象插件按照 SUT 可识别的格式加载到 SUT 中,从而完成测试用例加载过程;SUT 在接收到测试激励后的各种反馈信息,测试环境采用逆向过程进行接收。

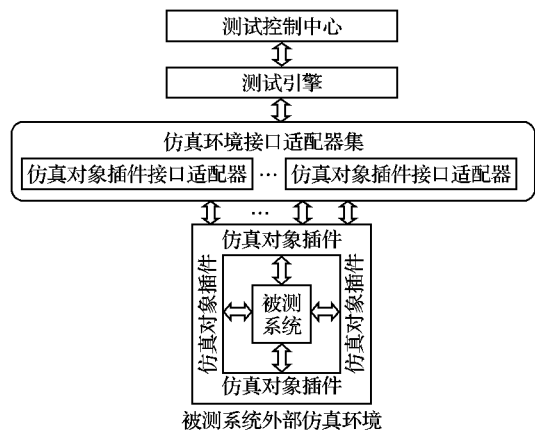


图4 基于 SED\_SCS\_STL 的通用自动化仿真测试环境架构

基于上述架构,测试控制中心、测试引擎独立于特定的被测系统实现,对不同的被测安全苛求软件系统,整个自动化仿真测试环境中只需要挂载不同的仿真对象插件,重新配置仿真环境接口适配器集,从而确保了测试环境的通用性。

## 4 实例应用

### 4.1 高速铁路 CTCS-2 级车站列控中心简介

伴随200km/h~250km/h和300km/h~350km/h客运专线的建设,我国铁路建设进入了一个蓬勃发展的新时期。我国目前采用 CTCS-2 级列车运行控制系统实现高速铁路运行的高效与安全。作为客运专线 CTCS-2 级列控系统地面设备的核心,车站列控中心的成败攸关整个客运专线 CTCS-2 级列控系统,客运专线 CTCS-2 级车站列控中心被定义为 SIL-4 级安全苛求系统,在任何情况下均不能导致危险侧输出,应具备高安全、高可靠、高性能的特点,其系统结构如图5所示。

### 4.2 软件系统仿真测试平台

针对客运专线 CTCS-2 级车站列控中心架构及基于 SED\_SCS\_STL 的通用自动化仿真测试环境架构,TestPLT\_TCC 的架构如图6~9所示。

下面给出在实际测试过程中,针对 CTCS-2 车站列控中心典型功能——临时限速的测试给出 SED\_SCS\_STL 编写的测试脚本(部分)如下。

```
TestCtr_Main. Script;
* 测试主控模块脚本 *
Impot TccScenario. *;
Mod_MainControl TccTestMain
```

```

{ * 定义部分 *
SimObj_Train: Test_Train;
TestExecTranc: Test_ExecTranc;
Test_Train. ID = 1;
Test_Train. Name = 'A 型车';
Test_Train. Length = 100;
Test_Train. Type = 'A';
Test_Train. StPos = 'K727 + 000';
Test_Train. EdPos = 'K827 + 000';
Test_Train. CurPos = 'K727 + 000';
* 测试执行逻辑部分 *
Test_ExecTranc. Start();
While( Test_Train. CurPos < 'K827 + 000')
( Test_Train. Run();
    If( Test_Train. CurPos = 'K727 + 100')
    (
        Test_ExecTranc. Exec( TccScenario. TstSce_1);
    )
    If( Test_Train. CurPos = 'K727 + 100')
    (
        Test_ExecTranc. Exec( TccScenario. TstSce_2);
    )
    ...
)
Test_ExecTranc. Commit();
* 测试配置部分 *
...
}

```

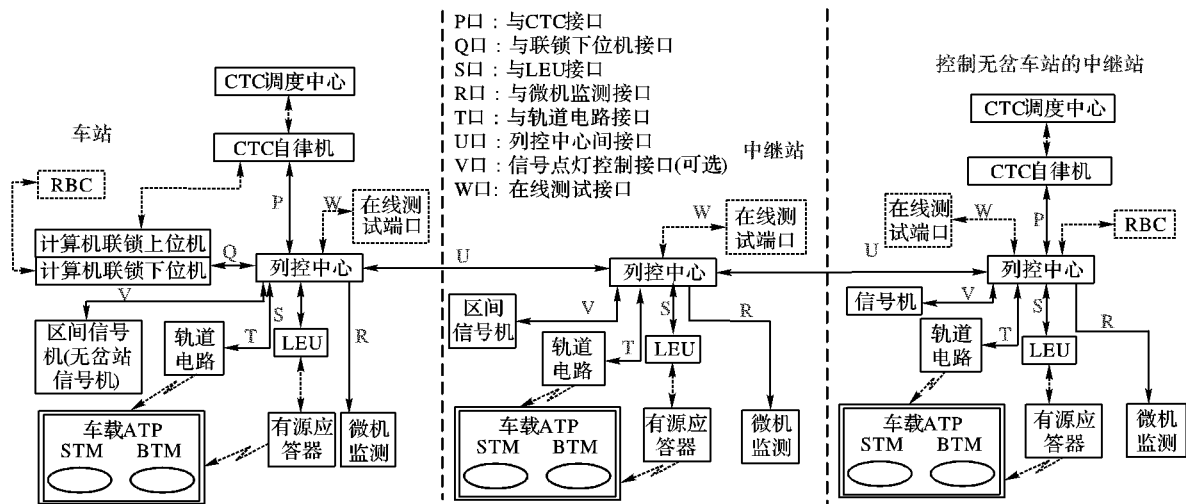


图5 客运专线 CTCs-2 级车站列控中心架构

TestExec\_Scenario. Script

```

* 测试执行模块脚本(场景级) *
Impot TccTestSeq. *;
Mod_SceExec TccScenario
{
SceObj: TstSce_1;
(
* 定义部分 *
SceExecTranc: Sce_ExecTranc;
TstSce_1. ID = 1;
TstSce_1. Name = '临时限速';
...
* 执行逻辑部分 *
Sce_ExecTranc. Exec ( TccTestSeq.
TstSeq);
)
...
SceObj: TstSce_2;
(
* 定义部分 *
SceExecTranc: Sce_ExecTranc;
TstSce_2. ID = 2;
TstSce_2. Name = '降级显示';
...
* 执行逻辑部分 *
Sce_ExecTranc. Exec ( TccTestSeq.
TstSeq);
)
}

```

TestExec\_Sequence. Script

```

* 测试执行模块脚本(测试序列级) *
Impot TccTestCase. *;

```

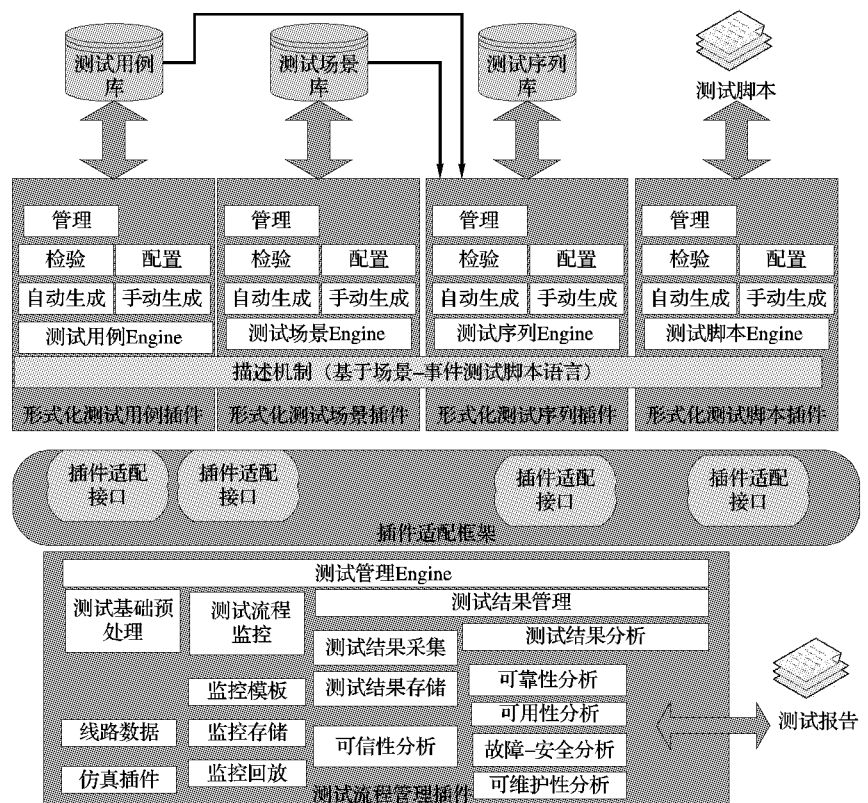


图6 TestPLT\_TCC 架构—测试控制中心

Mod\_SeqExec TccTestSeq

```

{ SeqObj: TstSeq_1;

```

```

(

```

```

* 定义部分 *

```

```

SeqExecTranc: Seq_ExecTranc;
TstSeq_1. ID = 1;
TstSeq_1. Count = 2;
...
* 执行逻辑部分 *
Seq_ExecTranc. Exec( TccTestCase. TestCase);
)

```

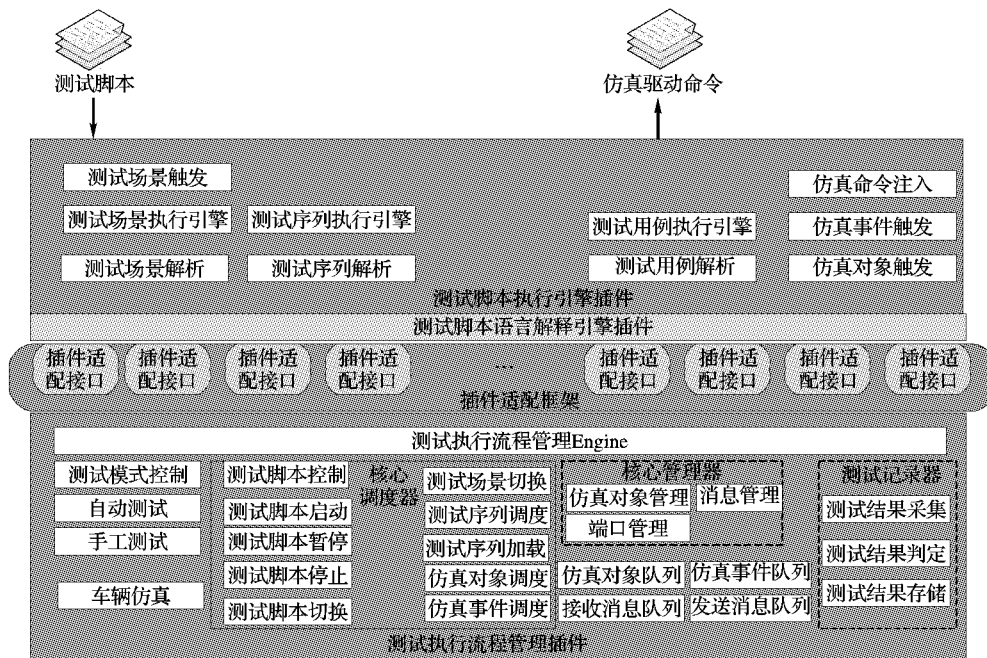


图7 TestPLT\_TCC 架构—测试引擎

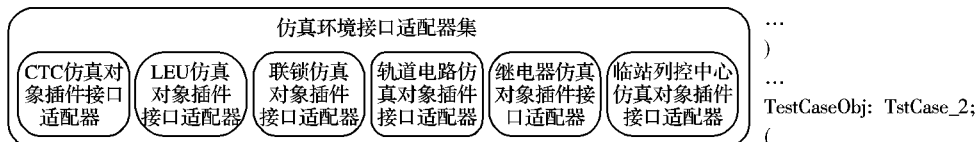


图8 TestPLT\_TCC 架构—仿真环境接口适配器集

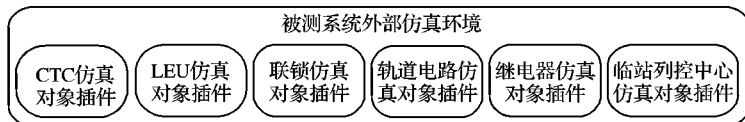


图9 TestPLT\_TCC 架构—被测系统外部仿真环境

TestExec\_Case. Script

\* 测试执行模块脚本(测试用例级) \*

Impot TccSimuObj. \*;

Mod\_CaseExec TccTestCase

{ TestCaseObj: TstCase\_1;

(

\* 定义部分 \*

TestData TstData;

TstCase \_1. ID = 1;

TstCase \_1. Name = '测试临时限速';

\* 测试数据部分 \*

TstData. Input = '40; 45; K737 + 100; 120';

TstCase \_1. Data = TstData;

\* 预期结果部分 \*

TstData. PreOutput = '0; 45; K737 + 100; 120';

...

\* 测试行为部分 \*

TccSimuObj. CTC. Send TstCase \_1. Data;

TccSimuObj. LS. Send TstCase \_1. Data;

TstData. RealOutput = TccSimuObj. CTC. Rev();

TstData. RealOutput = TccSimuObj. LS. Rev();

TstData. RealOutput = TccSimuObj. Leu. Rev();

TestExec\_SimObj. Script

\* 测试执行模块脚本(仿真对象级) \*

Mod\_SimObj TccSimuObj

{ SimObj CTC

{

Action: Send

(

Msg1

) On EventList( Eve1)

Action: Rev( Msg2)

Msg: Msg1

(

DataFormat

)

...

Event: Eve1

(

Name: Eve1;

Desc: ...

Condition: ...

)

}

...

SimObj LS

(...)

}

#### 4.3 基于 SED\_SCS\_STL 的作用于测试效果分析

在客运专线 CTCS-2 车站列控中心软件系统仿真测试中,本文提出的基于 SED\_SCS\_STL 的测试脚本技术能满足安全苛求软件测试的特殊要求,可实现对其闭环仿真测试。SED\_SCS\_STL 具有良好的测试描述能力,为客运专线 CTCS-2 车站列控中心软件系统仿真测试已建立了 5000 个测试场景,形成了 500000 个测试用例库及测试序列库,可针对客运专线 CTCS-2 车站列控中心软件系统的实际中可能出现的运营场景,组织各种不同类型的测试用例与测试序列并对测试反馈进行处理,基于 SED\_SCS\_STL 的编写的测试脚本执行效率高,中等规模的测试脚本(如上例)执行时间不超过 100 ms,定时精度可达 1 ms。在铁道部实施的客运专线 CTCS-2 车站列控中心产品的制式测试中,整个测试周期平均为 3 天,能够满足产品上线前的检测时间要求,获得铁道部的认可。

#### 5 结语

本文提出的针对安全苛求软件仿真测试的通用脚本语言 SED\_SCS\_STL 体现安全苛求软件测试的特点,能够对安全苛求软件测试进行通用化描述,包括安全性、安全防护、故障、测试反馈处理等方面。另外,通过针对该语言所设计的自动化仿真测试环境可完成与仿真模型、被测软件之间的交互,从而

实现安全苛求软件的闭环、仿真测试。基于 SED\_SCS\_STL 及其自动化测试环境已成功研发一套 CTCS-2 车站列控中心软件仿真测试平台产品,目前利用该产品已成功完成了铁道部要求的对 200 ~ 250 km/h 和 300 ~ 350 km/h 客运专线多个型号车站列控中心系统产品的制式测试和监督抽查工作。

在下一阶段,将进一步增强测试脚本语言的测试与测试控制描述能力,提高测试脚本的解析执行速度,实现从系统需求到测试脚本的自动生成。

#### 参考文献:

- [1] 吴芳美. 铁路安全软件测试评估[M]. 北京: 中国铁道出版社, 2001.
- [2] MOSLEY D J, POSEY B A. 软件测试自动化[M]. 北京: 机械工业出版社, 2003.
- [3] 王铁江. 安全关键软件的可靠性和安全性模糊评估及测试方法的优化[D]. 上海: 同济大学, 2003.
- [4] JURISTO N, MORENO A M, STRIGEL W. Software testing practices in industry [J]. IEEE Software, 2006, 23(4): 19-21.
- [5] KANGLIN L, MENGQI W. Effective Software Test Automation [M]. Alameda: Sybex, 2004.
- [6] PERRY W E. Effective methods for software testing [M]. New York: John Wiley & Sons, 2000.

## 中国计算机学会

### 全国第五届语义 Web 与本体论学术研讨会 (SWON2010) 征文通知

(2010 年 8 月 20—22 日 内蒙古大学, 呼和浩特)

目前在 Web 信息的表示和获取方式等方面存在着一些瓶颈,而语义 Web 的核心思想是通过增加一些语义信息,使得计算机能参与到自动处理 Web 信息的过程,并为实现智能化的 Web 应用提供必要的技术基础。语义 Web 吸取了人工智能、信息论、哲学、逻辑和计算复杂性等学科的研究成果,是多学科、多技术融合的产物。

全国语义 Web 与本体论学术研讨会 (SWON) 是中国计算机学会暨电子政务与办公自动化专委会主办的系列会议。SWON 2010 会议将于 2010 年 8 月在呼和浩特召开。会议目的是为学术界和工业界交流语义 Web 提供一个平台,反映国际国内关于语义 Web 的最新研究成果和进展。

会议录用论文中主要论文以英文方式由 IEEE Computer Society Press (EI 源刊) 正刊出版,其余论文将由核心期刊《计算机科学》专刊、《计算机与数字工程》正刊出版。会议期间除进行会议论文交流外,还将邀请著名专家做特邀报告,并继续评选大会优秀学生论文。

#### 一、征文范围(包括但不限于)

- 语义 Web 语言与工具
- 语义 Web 推理
- 语义信息标注
- 本体学习与元数据生成
- 语义 Web 知识表示与管理
- 语义 Web 服务
- 语义检索和查询
- 本体存储与管理
- 语义 Web 挖掘
- 语义 Web 安全
- 本体构建
- 语义集成和映射

#### 二、来稿要求

本次会议主要通过网上投稿,尽量不要通过 E-mail 投稿,拒收纸质稿件。严禁一稿多投。

中英文稿均可,一般不超过 6000 字,为了便于出版论文集,来稿必须附中英文摘要、关键词、资助基金与主要参考文献,注明作者及主要联系人姓名、工作单位、详细通信地址(包括 E-mail 地址)与作者简介。稿件要求采用 WORD 或 PDF 格式。

#### 三、联系信息

1. 投稿地址: <http://www.easychair.org/conferences/?conf=wisa2010>

投稿联系人: 华东师范大学 周傲英 周敏奇 (mqzhou@sei.ecnu.edu.cn)

2. 大会网站: [www.neu.edu.cn/wisa2010](http://www.neu.edu.cn/wisa2010)

3. 会务情况: 内蒙古大学 高光来 周建涛 (cszjtao@imu.edu.cn, wisa2010@imu.edu.cn)

#### 四、重要日期

1. 征文截止日期: 2010 年 3 月 31 日

2. 录用通知发出日期: 2010 年 4 月 30 日

3. 正式论文提交截止日期: 2010 年 5 月 20 日

4. 会议召开日期: 2010 年 8 月 20—22 日