

基于 Web 的多领域可视化建模系统设计与实现

赵顺华 吴义忠 沈 博

(华中科技大学国家 CAD 支撑软件工程技术研究中心 武汉 430074)

摘 要 针对单机环境下基于 Modelica 语言的多领域建模与仿真软件不利于知识共享和积累、软件维护升级困难等不足,研究了 B/S 架构的多领域建模与仿真技术,设计并开发实现了基于 Web 的多领域建模系统 WebMWorks;采用 Silverlight 技术和 WCF 技术等,实现了系统浏览器端的可视化及其与服务端程序的通信。最后,给出了系统运行实例。

关键词 Web,多领域建模与仿真,可视化建模,Modelica

中图分类号 TP311,TP391 **文献标识码** A

Design and Implementation of Multi-domain Visual Modeling System Based on Web

ZHAO Shun-hua WU Yi-zhong SHEN Bo

(National Engineering Research Center of CAD Information Technology, Huazhong University of Science and Technology, Wuhan 430074, China)

Abstract Modelica-based multi-domain modeling and simulation software running on desktop has many disadvantages over Web-based software, which goes against the share and deposit of knowledge, and its maintenance and updating is difficult. Multi-domain modeling and simulation technology based on B/S architecture was researched, and a multi-domain modeling and simulation system—WebMWorks, which is based on Web, was designed and implemented. WebMWorks adopts silverlight and WCF technology to implement the visualization of models on browser side, and communication between browser and Web server. Finally a modeling instance on the WebMWorks system was given.

Keywords Web, Multi-domain modeling and simulation, Visual modeling, Modelica

1 引言

Modelica^[1]是一种开放的、面向对象的多领域建模语言。它以方程为基础,可跨越不同领域(如机械、电子、电力、液压、热、控制等等),方便地实现复杂物理系统的统一建模。该语言的出现,很好地解决了不同领域模型之间进行无缝集成和能量传输的问题。目前,支持 Modelica 语言较成熟的商用软件主要有 Dymola, MathModelica, OpenModelica, MWorks 等^[2]。其中,Dymola^[3]是第一个支持 Modelica 语言的建模工具,由 Modelica 语言创始人 Hilding Elmqvist 开发。而 MWorks^[4,5]是华中科技大学 CAD 中心历时 8 年开发出来的支持 Modelica 语言的多领域建模平台,填补了中国在多领域建模领域的空白。随着 Modelica 语言的推广和普及,越来越多的企业或行业开始引入基于 Modelica 语言的仿真平台进行复杂系统建模和仿真分析^[6-11]。

但是,单机环境下支持 Modelica 语言的建模仿真软件,存在建模与求解模块耦合性太强、涵盖知识的模型库共享性差、系统维护升级困难等问题。单机环境就好像一个信息孤岛,不利于行业设计模型知识的共享和积累,而且模型知识容易流失。而基于 Web 的建模环境可以很好地解决上述问题,因此本项目以 MWorks 作为后台服务器,开发基于 Web 的

领域建模环境,实现多领域物理系统前端建模、后端求解、结果后处理以及模型管理的一体化。目前,基于 Modelica 语言的 Web 系统已见诸于相关文献,如 OMWeb^[12],它是一种基于 Web 的教学环境,用于 Modelica 语言的编程教学。OMWeb 提供了文本编程环境,能够编写基于 Modelica 语言的代码。另外,文献[13]介绍了网络化的电气工程虚拟实验室 iEEVL, iEEVL 能编辑设计实验及求解特定实验。文献[14]介绍了基于网络的仿真采用的框架结构以及技术手段,并提供了原型。与前面的类似,文献[15]采用 OpenModelica 作为编译器,创建了 Web 下的仿真环境,它是一个能够选择实验的虚拟实验室,主要应用于网络教学。然而这些基于 Modelica 的 Web 建模系统相比单机环境下的建模仿真软件,功能和实用性不强,它们缺乏基本的可视化建模能力,也没有充分利用 Modelica 提供的资源及网络化优势。为了充分发挥网络的优势,便于建模知识的交流和分享,开发基于 Web 的多领域可视化建模系统势在必行。

2 系统体系结构

图 1 所示为 WebMWorks 系统体系结构图,其主要由浏览器端、服务端和数据端 3 部分组成。浏览器端是建模界面和后处理界面,主要用于用户的前端建模和查看结果;服务器

收稿日期:2012-08-11 返修日期:2012-11-09 本文受国家自然科学基金项目(51175198)资助。

赵顺华(1987-),男,硕士生,主要研究方向为多领域仿真技术,E-mail:hust_2012@126.com;吴义忠 男,教授,博士生导师,E-mail:cad.wyz@mail.hust.edu.cn(通信作者)。

端完成模型的编译求解,以及处理浏览器端的请求,响应并发送相应的文件;数据端用于管理用户信息和模型信息,主要包括用户个人资料以及模型图形信息、模型仿真结果等。

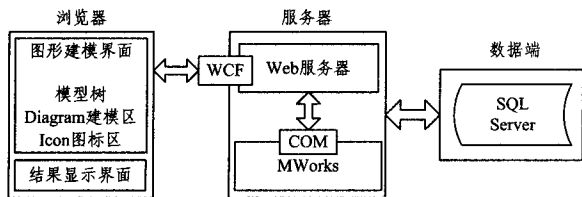


图1 WebMWorks系统框架结构图

2.1 浏览器

图形建模界面以及计算结果显示在客户端浏览器上,它以单机版 MWorks 系统为原型,主要功能包括:

- 1)模型库管理:Modelica 标准模型库、开发库,以及用户个人库的模型树状浏览与管理;
- 2)模型文本显示:mo 是 Modelica 表示的模型的全部信息,WebMWorks 提供了模型信息浏览功能,可以查看模型的 mo 文本;
- 3)图标编辑:Modelica 语言图标编辑;创建模型图标;
- 4)组件建模:Diagram 组件建模;组件连接;模型编辑;
- 5)模型说明:查看 HTML 说明视图;
- 6)模型仿真结果显示:图形显示模型仿真结果;
- 7)参数编辑:修改模型组件的参数。

本文重点介绍可视化建模部分,即图标建模和组件建模。

2.2 服务端

服务器端主要包括两个部分:Web 服务器和计算服务器。

Web 服务器用来响应浏览器的请求,处理请求并发送文件到浏览器,供浏览器显示。本系统采用的是 IIS 服务器,采用 WCF 实现。

计算服务器采用本单位的 MWorks 软件,以 COM 接口提供各种建模、编译和求解命令,其结果通过 Web 服务返回给客户端。

2.3 数据端

数据库/文件仓用来管理用户信息和模型信息:①将多领域模型库文件(MO 文件)严格按照 Modelica 规范约定的目录形式组织存储于文件仓中,同时映射入数据库;②将所有模型的图形化信息批量导入数据库进行管理,以实现模型库的快速加载和图形化显示;③文件仓同时以二进制文件的形式储存模型的仿真结果,同时映射入数据库;④数据库同时管理用户的个人资料等信息。

3 系统关键技术

3.1 WCF 服务

WCF(Windows Communication Foundation)是由微软开发的一组数据通信的应用程序接口,为 .NET 框架的一部分。与 Webservice 相同,它可以为跨平台应用提供网络服务,同时 WCF 还为应用程序之间的通信建立了基础结构,可以把原来的、现在的、将来出现的不同的通信方式整合在一起,提供一种统一的编程模型。所有 WCF 服务的连接都是由客户

端的服务代理(WCF Service Proxy)来运行,开发人员不用花费太多心思在通信上,只需要在代码中取用 WCF Service Proxy 对象即可。

WCF 在 WebMWorks 系统中起到桥梁的作用,连接浏览器端与服务端,保证浏览器端与服务端的通信。由于建模过程在浏览器端,而编译求解过程在服务端,为了保证浏览器端与服务端模型信息的一致,必然要做到同步模型。所谓同步模型,就是服务端模型的组件信息、图形信息、连接信息及参数信息等通过与前端通信,来与浏览器端的模型信息保持一致。在 WebMWorks 系统中,建模信息、编译求解命令、仿真结果的传输都采用了 WCF,WCF 保证了建模与仿真过程中模型信息的同步。

3.2 数据交换

Modelica 模型信息主要包括模型的变量(包括参数)信息、方程信息以及图形信息^[5],这些信息均存储在 mo 文本文件中。在 MWorks 中,提供了文本编辑和可视化建模(图标 Icon 编辑和组件 Diagram 拖放及连接)等功能,以实现对模型的编辑。在 WebMWorks 中,浏览器端的可视化信息(Icon 和 Diagram)是由后台 MWorks 同步模型的相关信息通过 WCF 服务获取并显示的。

• SVG 格式

SVG 是一种基于可扩展标记语言 XML,用于描述二维矢量图形的图形格式。

它采用文本描述图形对象,具有交互性和动态性,而且完全支持 DOM(文档对象模型),包含图形信息的 svg 文件比较小,适合网络传输。WebMWorks 系统中,组件的图形信息、参数信息等采用 SVG 格式传输。

• GLB 格式

glb 文件用二进制方式保存模型的组件信息、组件图形与位置信息、组件之间连接线的信息,采用二进制流方式的 glb 文件比较小,适合网络传输。WebMWorks 系统中,服务器端保存的模型均采用 GLB 格式。浏览器以及 Web 服务器之间是通过交换包含模型图形化信息或者模型的参数信息的 svg 文件、glb 文件实现通信的。

3.3 Silverlight 技术

Silverlight 是一种跨浏览器、跨客户平台的技术,能够设计、开发和发布有多媒体体验与富交互(RIA, Rich Interface Application)的网络交互程序。其包含了 WPF 的技术特性,具有优越的矢量图形、动画和多媒体支持能力,内置丰富的网络通信功能,提供了基于 HTTP 的跨域网络访问支持。Silverlight 应用程序能访问 WCF、SOAP 和 ASP.NET AJAX 服务,并能接收 XML、JSON 和 RSS 等格式的数据^[16]。

• 矢量图形支持

Silverlight 提供了强大的二维绘图功能,能绘制直线、矩形、椭圆、多边形等。Silverlight 将图形绘制所需的功能集成到 shape 类以及 Geometry 类,shape 类使用方便、简单,能绘制一些基本的图形。Silverlight 提供了一个功能最强的 Path 类,它不仅能绘制简单图形、组合图形,还能绘制更加复杂的元素,比如曲线等。Path 类包含一个简单的 Data 属性,Data 属性接受定义形状或路径的 Geometry 对象。应用以上两种

类可以方便地绘制出所需图形。

• 图形变换

Silverlight 完善的图形变换功能使得绘制图形变得容易。Silverlight 提供的常用的图形变换功能有: TranslateTransform(平移变换)、RotateTransform(旋转变换)、MatrixTransform(矩阵变换)以及 TransformGroup(组合变换)^[17]。

4 系统实现

WebMWorks 系统采用 Silverlight 提供的绘图功能完成模型图形的绘制,下面重点介绍可视化建模部分的实现,包括图标建模、拖放式组件建模及组件连接。

4.1 图标建模

图标是 Modelica 模型作为组件嵌入到其它模型时显示的图形。图标视图提供了图形创建和编辑功能,能够创建以及编辑包括直线、矩形、椭圆、多边形、文本和图片在内的 Modelica 中的常见图形。

在 WebMWorks 系统中,图标建模是通过图 2 所示的类系完成的。

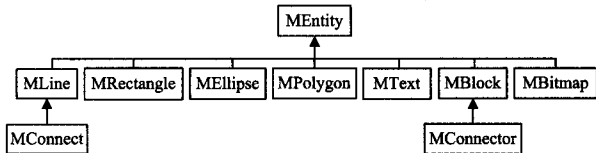


图 2 实体类继承关系

图 2 中, MEntity 是图形实体类的基类,其中定义了所有图形的基本属性以及常用方法。当用户打开已有模型时, WebMWorks 系统调用 WCF 服务,根据模型全名从服务端找到相应 GLB 文件,接着服务端把文件内容以字符形式传送到浏览器。浏览器端用 BinaryReader 对象加载包含模型的图形信息的文本内容,接着从 Binary Reader 对象中读取图形实体类

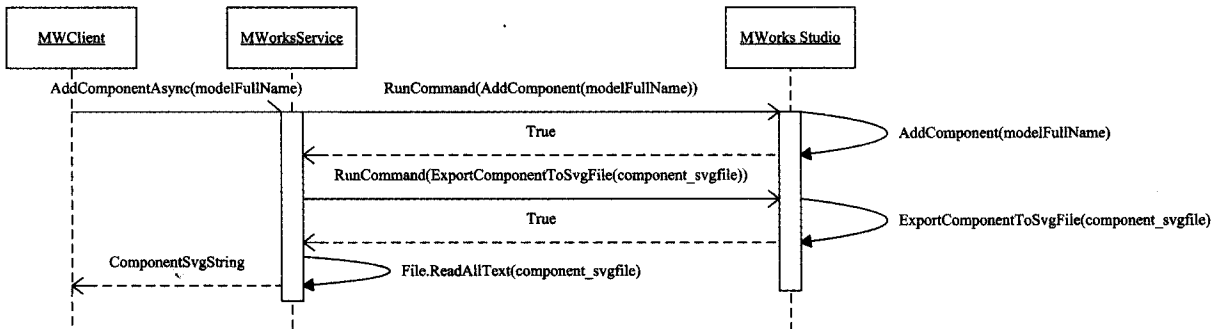


图 4 组件图形信息产生过程

MWClient 位于浏览器客户端,用户拖拽建模时,系统调用 WCF 服务中 AddComponentAsync() 命令,服务器端调用 MWorks Studio COM 接口中添加组件命令,在当前模型中添加相应组件。成功添加组件后,接着调用 MWorks 命令,将组件的图形信息导出到服务端文件系统的 svg 文件中,接着 Web 服务器程序读取 svg 文件,将其中信息以字符串形式返回到客户端。

• 组件对象设计

组件图形部分为 MBlock 的实例, MBlock 的字段和事件如图 5 所示。

参照 MWorks 中组件的功能,本文系统中的组件首先能

型以及图形属性,然后从浏览器端创建相应图形对象并加入到实体链表中。此时,显示图标只需要调用实体链表内各个实体的 Draw 函数即可完成。具体 GLB 解析算法如图 3 所示。

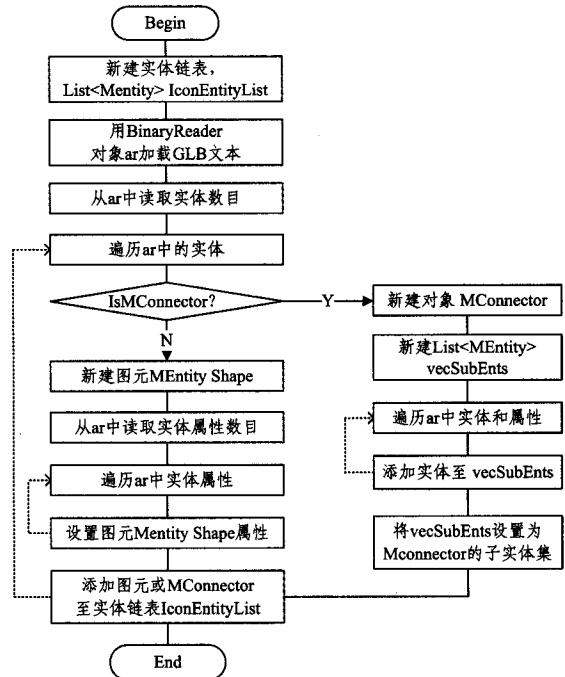


图 3 GLB 解析算法

4.2 组件生成

组件是 Modelica 语言中模型的实例,是图形化实体的集合,图形化建模过程就是选择相应组件并连接组件的过程。

• 组件拖放

图 4 所示为在系统中添加组件时,服务器、客户端以及 Mworks 之间的通信关系。

够显示,且具有平移、旋转等图形变换和响应鼠标键盘事件的功能。组件内包含子元素,因此组件应该具有容器的性能。而 MBlock 的 Canvas 字段、mt 字段、m_vecClassEntities 字段以及事件 MouseLeftButtonDown 可以满足组件的上述功能。Canvas 是 silverlight 提供的可视化容器类,可以显示常见图形,并且允许 canvas 嵌套,mt 代表矩阵变换,能够进行各种复杂的图形变换。m_vecClassEntities 是 List<MEntity> 的对象,用于记录组件的子实体集,而事件 MouseLeftButtonDown 响应组件的鼠标动作,如鼠标单击、双击等,鼠标单击可以选中组件;双击组件,弹出组件参数界面,可以修改参数。

```

Class
  MBlock
  字段:
    Canvas
    IsPicked
    m_dClassCoordsys
    m_vecClassEntities
    mt
    mx
  事件:
    MouseLeftButtonDown

```

图5 MBlock 字段、事件

• 组件图形显示

组件的图形信息包含在 svg 文件中,因此只要将 svg 文件中的图形信息解析出来,并调用实体的 Draw 命令,就显示出了组件。WebMWorks 提供了解析 svg 文件所需的图形类,图6展示了 svg 图形类之间的关系,SVGElement 是其他类的基类,它提供了基本的属性和方法。当包含 svg 信息的字符串传入到解析函数时,整个 svg 字符串当作一个 svgElement,然后遍历 svgRoot 子元素。如果图形类是 SVGGroup 类型,则新建 MConnector 对象,设置 MConnector 的缩放系数,遍历 MGroup 的子实体集,添加实体到 MConnector 对象的实体链表中;如果图形类型不是 MGroup,新建相应的图形对象,将新建的图形对象添加到实体链表中。具体 svg 解析算法如图7所示。

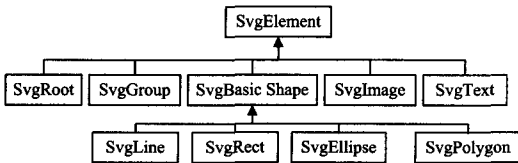


图6 svg 图形类

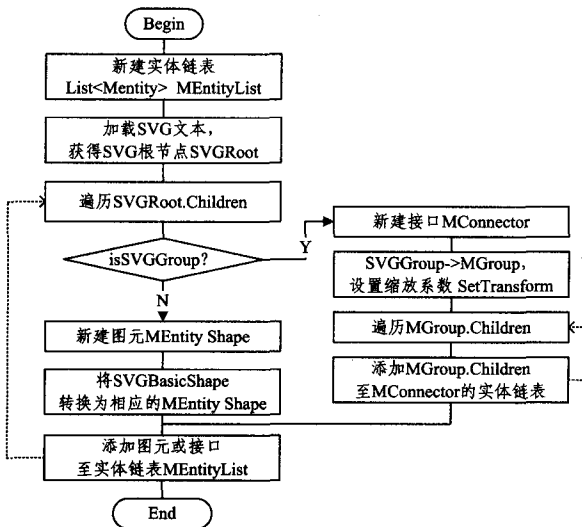


图7 svg 解析算法

4.3 组件连接

连接是 Modelica 连接方程(connect)的图形表示,每个连接对应一个 connect 语句,连接的图形信息记录在 connect 语句的注解(annotation)中,连接表示组件之间的相互联系,组件包含连接器(connector)信息,它是模型与外界通信的接口。组件进行连接时,首先判断连接点处是否有接口即 Connector

对象,而后选择路径创建连接,直到点中另外一个接口,连接创建完成。

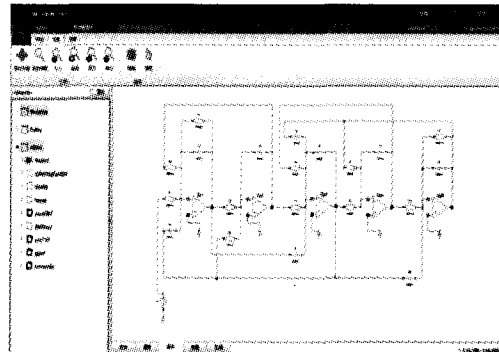
为了在浏览器端实现上述功能,本文提供 MConnect 类表示连接,MConnector 表示连接器,如图2所示,MConnect 继承自 MLine,可以用直线图形化表示连接,同时还记录相互连接的接口 BeginMConnector 和 EndMConnector。

连接器 MConnector 继承自 MBlock,除了包含 MBlock 的字段和属性外,还包括 List<MConnect> 类型的 Connections,其表示与该连接器相连的连接,以及 MBlock 类型的 parentBlock,其代表连接器属于哪个组件。

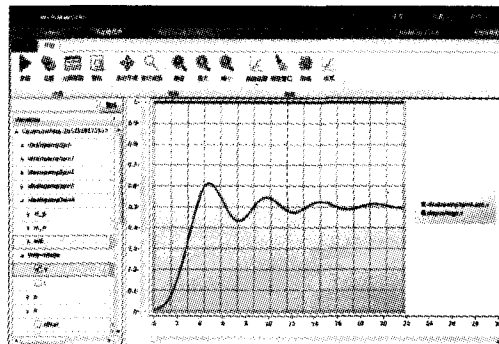
有了 MConnector 以及 MConnect,下一步任务就是连线,在进行连线的过程中,连接线永远以与坐标系正交的方式显示。为实现此需求,制定了创建连接线的正交化算法:①在连线模式下,鼠标首次在组件的连接器点击,新建一条连线,并设置此连接点为连线的起始端口;②鼠标在 Diagram 视图区域中空白位置点击,则将此位置添加至连线的中间节点链表,然后绘制中间线;③直到鼠标在另外一连接器点击或者点击右键,分别转向步骤④或步骤⑤;④鼠标在另外一连接器点击,则检查连线是否符合连线规范,如符合,将此连接器设置为连线的终点端口,并将连线添加至两端连接器的连线链表中,连线完成;否则,弹出错误警告;⑤连线过程中点击右键,则弹出右键菜单,根据菜单选项进行下一步操作:取消连线或创建连接器。

5 系统运行实例

图8所示为 WebMWorks 系统建模实例,这是带有运算放大器的低通滤波器。



(a) 模型组件视图



(b) 结果显示

图8 WebMWorks 系统建模实例及结果显示

图8(a)为模型组件视图,为模型提供可视化环境。通过 (下转第159页)

假混沌不透明谓词,它们将打印对应的混沌不透明谓词的输出,并以此方式统计改变密钥之后有效的不透明谓词的个数。结果如表4所列。

表4 改变密钥前后有效混沌不透明谓词的个数

测试程序	永真		永假	
	更改之前	更改之后	更改之前	更改之后
九宫格程序	5	0	5	5
哈夫曼解压缩程序	5	0	5	5
DES加密程序	5	0	5	5

从实验的结果可以看出,永真的混沌不透明谓词在密钥修改之后几乎全部失效,说明新的构造方法对密钥是敏感的,密码安全性较高。建议在混淆代码的过程中多使用永真的混沌不透明谓词。

结束语 本文研究了基于混沌理论的不透明谓词在代码混淆中的应用,对利用混沌不透明谓词技术混淆代码的整个过程进行了比较详细的描述。并通过实验从混淆转换前后程序的复杂度变化、程序控制流的复杂度变化两方面评价了混淆转换的有效性。对整个混淆转换系统的安全性进行了分析,并通过实验验证了整个混淆转换系统对密钥的高度敏感性,说明其密码安全性较强。如何通过程序来辅助用户做出最佳的混淆策略以达到运行效率和安全性的最佳平衡是下一步研究的方向。

(上接第141页)

组件视图,可以方便地观察模型的组件以及组件关系。图8(b)为结果显示界面,显示输入信号和输出信号之间的关系,对于本实例,输入信号是指阶跃信号,输出是最右边运算放大器的输出信号,信号数据分别用蓝色线和红色线表示。

结束语 本文设计并实现了一个基于B/S结构的多领域可视化建模系统WebMWorks,并提供了建模实例。相比单机环境下的建模系统,本系统具有以下特点:易于维护,方便升级部署;解决了可移动性和共享性差的问题;减少了客户端的压力;具有分布式、异地化的特性。

参考文献

[1] Modelica and the Modelica Association [EB/OL]. <http://www.modelica.org>, 2012-03-05

[2] 吴义忠,陈立平. 多领域物理系统的仿真优化方法[M]. 北京:科学出版社, 2010:7-8

[3] 维基百科 Dymola [EB/OL]. <http://en.wikipedia.org/wiki/Dymola>, 2012-06-30

[4] 武汉天喻软件有限公司. 多领域物理系统建模与仿真平台MWorks [EB/OL]. <http://www.hustcad.com/NewContent.asp?ID=520>, 2012-07-01

[5] 吴义忠,刘敏,陈立平. 多领域物理系统混合建模平台开发[J]. 计算机辅助设计与图形学学报, 2006, 18(1): 120-124

[6] 秦新燕. Modelica语言在电路建模与仿真中的应用[J]. 湖北教育学报, 2007, 24(8)

[7] 莫雨峰,刘成良. 基于Modelica的航空发动机电气系统仿真分析研究[D]. 上海:上海交通大学, 2004

参考文献

[1] 楼俊钢,江建慧,帅春燕,等. 软件可靠性模型研究进展[J]. 计算机学报, 2010, 37(9): 13-17

[2] 史扬,曹立明,王小平. 混淆算法研究综述[J]. 同济大学学报, 2005, 33(6): 813-819

[3] 付剑晶. 代码干扰变换在软件保护中的使用[J]. 计算机应用与软件, 2008, 25(4): 103-105

[4] 贾春福,王志,刘昕,等. 路径模糊:一种有效抵抗符号执行的二进制混淆技术[J]. 计算机研究与发展, 2011, 48(11): 2111-2119

[5] 袁征,冯雁,温巧燕,等. 构造一种新的混淆Java程序的不透明谓词[J]. 北京邮电大学学报, 2007, 30(6): 103-106

[6] Yuan Zheng, Wen Qiao-yan, Wu Wen-ling, et al. An ID-based watermarking scheme for java programs[C]//EUC Workshops, 2006. Berlin: Springer-Verlag, 2006: 848-857

[7] 袁春,钟玉琢,贺玉文. 基于混沌的视频流选择加密算法[J]. 计算机学报, 2004, 27(2): 257-262

[8] Cormen T H, Leiserson C E, Rivest R L, et al. Introduction to Algorithms [M]. The MIT Press, 2009

[9] 赵玉洁,汤战勇,王妮,等. 代码混淆算法有效性评估[J]. 软件学报, 2012, 23(3): 700-711

[10] Collberg C. A Tool for the Study of Software Protection Algorithms [EB/OL]. <http://sandmark.cs.arizona.edu>, 2012-05-24

[8] 董怡文,赵翼翔,陈新度. 基于Modelica的硫化机液压传动系统仿真[J]. 机床与液压, 2010, 38(17): 114-117

[9] 于红. 基于Modelica的球杆平衡系统建模仿真[J]. 机械工程与自动化, 2011(1): 128-130

[10] 陈卫平,王波兴,黄运保. 基于多领域统一建模飞机起落架系统的仿真与分析[D]. 武汉:华中科技大学, 2009

[11] 百度百科-Modelica [EB/OL]. <http://baike.baidu.com/view/5623020.htm>, 2012-01-10

[12] Torabzadeh-Tari M, Hossain Z M, Fritzson P. OMWeb-Virtual Web-based Remote Laboratory for Modelica in Engineering Courses [A] // Proceedings 8th International Modelica Conference, 2011 [C]. Dresden, Germany: Linköping University Electronic Press, 2011: 153-159

[13] Shi Zheng-yin, Zhao Sheng-lin, Zhu Shan-an. An Internet-based Electrical Engineering Virtual Lab-Using Modelica for Unified Modeling [A] // 2011 IEEE 3rd International Conference on Communication Software and Networks (ICCSN 2011), 2011 [C]. Xi'an, China: IEEE, 2011: 555-559

[14] Eissen S M, Stein B. Realization of Web-based simulation services[J]. Computers in Industry, 2006, 57(3): 261-271

[15] Duarte O. UN-VirtualLab: A Web simulation environment of OpenModelica models for educational purposes [A] // Proceedings 8th International Modelica Conference, 2011 [C]. Dresden, Germany: Linköping University Electronic Press, 2011: 773-778

[16] 杨雷,邵城. Web环境下的Simulink仿真系统研究与开发[D]. 大连:大连理工大学, 2010

[17] MacDonald M. Pro Silverlight 4 in C# [M]. New York: Apress, 2010: 260-278, 310-311