

文档格式转换自动化

测试技术与方法

牟永敏 著



清华大学出版社

文档格式转换自动化 测试技术与方法

牟永敏 著

清华大学出版社
北京

内 容 简 介

本书介绍“文档格式转换测试”的相关概念、技术和方法,结合 OOXML/UOF 文档转换项目的测试实践,提出了从分析文档底层的 XML 代码入手,分析提取两种不同文档格式下对应 XML 的实现代码中关于文档内容的属性,通过分析和对比这些不同标准下的属性,实现不同格式文档之间转化的自动化测试,包括文字、表格和演示文稿等内容的自动化测试技术和方法。

本书可以供从事软件测试研究的高校教师、针对不同文档格式和标准转化开发和测试的企业工程师以及软件测试高层管理者参考,也可作为高等院校计算机相关专业研究生的教材和参考书。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

文档格式转换自动化测试技术与方法/牟永敏著.--北京:清华大学出版社,2014
ISBN 978-7-302-36829-8

I. ①文… II. ①牟… III. ①软件—测试 IV. ①TP311.5

中国版本图书馆 CIP 数据核字(2014)第 127815 号

责任编辑:付弘宇 薛 阳

封面设计:迷底书装

责任校对:焦丽丽

责任印制:刘海龙

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦 A 座

邮 编:100084

社 总 机:010-62770175

邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质量反馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

印 装 者:北京密云胶印厂

经 销:全国新华书店

开 本:185mm×260mm 印 张:16

字 数:390 千字

版 次:2014 年 8 月第 1 版

印 次:2014 年 8 月第 1 次印刷

印 数:1~1000

定 价:69.00 元

产品编号:057602-01

前 言

不同的国家、不同的民族都在使用着不同的办公软件。随着无纸化办公时代的到来,办公软件已经成为人们日常办公、学习所必备的应用软件之一。日常办公常用的软件主要包括文字处理(Word)、电子表格(Spreadsheet)以及演示文稿(Presentation)等,因其强大的文档编辑功能,不管在政府部门、企业还是个人应用中,办公文档都占据着极为重要的地位。这就要求一种办公文档格式不仅能够被不同办公软件理解和应用,而且也要求不同的文档格式之间能够相互转换。

目前,文档格式转换的测试主要是人工测试,测试内容包括文档格式在功能、性能、系统等方面是否转换正常。但是办公文档的功能点繁多,以文字处理为例,仅常用功能就已达200多个,对每个功能点需要根据其属性范围或标准规定的枚举值设计不同的测试用例,至少需要800个测试用例,另外,需要考虑文档格式的双向转换,测试用例约合1600多个。测试用例的设计工作繁杂,同时,人工测试是靠人眼对比两种不同环境、不同标准下的文档转化后的效果,工作量很大、内容繁杂,测试过程中难免出现人为的误差。其次,办公软件不能完全地支持标准,人工测试有一定的局限性,存在测试盲区。长期以来,人们对文档格式转换的自动化测试技术和方法进行了不断研究,有人提出用QTP等自动化测试工具,录制办公软件界面上按钮的操作,生成测试脚本,参数化测试输入,通过软件界面的捕获回放来生成测试用例。还有人提出截取两种不同文档的屏幕效果图,通过分析两种效果图,对比两种文档的转化效果是否正常。

通过长期的文档格式转化测试实践,使我们认识到,通过分析文档底层XML的实现代码,提取两个不同文档对应的XML代码中文档格式标准描述信息,建立文档格式统一模型,从而提出了一种文档格式转换自动化测试技术和方法,解决了人工测试中的盲区、办公软件不能很好地支持标准等问题,有效提高了测试的效率和准确性,为文档格式和标准转化自动化测试提供了一条崭新的途径,也为各种不同展示效果之间的自动化测试奠定了基础。

全书共分7章。第1章主要介绍文档格式转化的研究现状;第2章以项目组的OOXML/UOF文档转换器的测试项目为基础,介绍基本的测试理论以及文档格式转换的测试流程和测试特性;第4章以办公文档的电子表格为例介绍测试用例自动生成的技术方案;第5章介绍建立文档格式之间的功能点中间模型和自动化提取文档待测功能点底层XML代码中的原始属性集合的技术和方法;第6章介绍测试过程中的用例管理和缺陷管理方案,对这种测试策略下的覆盖率和测试用例关联的相关性进行了分析;第7章阐述软

件自动化测试的相关概念并介绍几种自动化测试工具。

全书由牟永敏、王锐、董伟娜、齐胜、丁媛、朱绪利、李良杰、刘梦婷、张亚楠、李慧丽、刘昂、杨志嘉、白倩倩、范浩杰、姜志超、王军合著而成。最后由牟永敏统一修改定稿。由于时间仓促,作者水平有限,书中欠妥和纰漏之处在所难免,恳请读者和同行批评指正。

编 者

2014年4月

目 录

第 1 章 绪论	1
1.1 背景介绍	1
1.2 文档格式转换研究现状	2
1.2.1 主流文档格式转换器	2
1.2.2 文档格式转换开发技术研究现状	3
1.2.3 文档格式转换测试技术研究现状	4
1.3 技术分析	5
1.4 专著内容	6
第 2 章 文档格式转换测试	8
2.1 文档格式转换测试过程概述	8
2.2 功能测试	9
2.2.1 黑盒测试的方法	10
2.2.2 文档格式转换功能测试	13
2.3 集成测试	23
2.3.1 集成测试概述	23
2.3.2 集成测试策略	24
2.3.3 文档格式转换集成测试	26
2.4 互操作测试	29
2.5 性能测试	30
2.5.1 性能测试概述	30
2.5.2 文档格式转换性能测试	31
2.6 系统测试	32
2.6.1 系统测试概述	32
2.6.2 系统测试过程	33
2.6.3 文档格式转换系统测试	35
2.7 回归测试	38
2.7.1 回归测试概述	38
2.7.2 回归测试策略和用例维护	39
2.7.3 文档格式转换回归测试	41
小结	41
第 3 章 文档处理关键技术	42
3.1 文档格式概述	42

3.2 XML 验证技术 Schema	48
3.2.1 概念与语法	48
3.2.2 UOF 2.0 Schema 格式	49
3.3 查询技术 XPath	50
3.3.1 XPath 的概念	50
3.3.2 XPath 语法简介	51
3.4 办公软件提供的 SDK 技术	53
3.4.1 Open XML SDK	54
3.4.2 UOF Open SDK	54
小结	56
第 4 章 测试用例自动生成	57
4.1 文档格式转换人工测试用例的设计	57
4.2 测试用例生成方法概述	60
4.3 测试功能点需求模型	62
4.3.1 文档格式标准形式化定义	62
4.3.2 测试功能点相关定义	64
4.3.3 测试功能点实例	64
4.4 测试功能点组合覆盖	65
4.4.1 组合覆盖分析法选择测试数据	65
4.4.2 测试功能点组合覆盖实例	66
4.5 文档格式标准建模	67
4.5.1 UML 类模型	67
4.5.2 UML 与 XSD 的层次关系	70
4.5.3 基于 XSD 的 UOF 2.0 标准分析	71
4.5.4 UML 与 XSD 组件之间的映射关系	73
4.6 测试用例自动生成的系统架构	79
4.7 测试用例自动生成的实现	80
4.7.1 模型层实现方案	81
4.7.2 对象层实现方案	82
4.7.3 实例文档层实现方案	87
4.8 应用与分析	89
4.8.1 原型系统视图	89
4.8.2 应用分析	91
小结	94
第 5 章 功能点自动对比可视化模型	95
5.1 功能点对比目的	95
5.2 UOF 和 OOXML 格式待测功能点分析	96

5.3	功能点对比的可行性分析	99
5.4	文档格式转换自动对比测试的技术路线	101
5.4.1	文档测试用例自动对比思路分析	101
5.4.2	文档格式转化自动化测试系统架构	102
5.5	文档格式转换测试用例自动对比的实现	103
5.5.1	基于功能点的对象模型	104
5.5.2	基于文档的对象模型	109
5.5.3	转换器自动化测试工具的对象模型	110
5.6	自动对比流程	111
5.7	待测功能点的自动化测试的实例分析	113
5.7.1	段落模块	113
5.7.2	元数据模块	115
5.8	应用与分析	117
5.8.1	应用举例	117
5.8.2	结果分析	119
	小结	120
第6章	文档格式转换自动化测试过程管理	121
6.1	人工测试过程管理	121
6.1.1	人工测试过程中的测试用例管理	121
6.1.2	人工测试过程中的缺陷管理	121
6.2	自动化测试过程管理	125
6.3	功能点路径	127
6.4	覆盖率度量	130
6.5	关联度分析	132
6.5.1	关联度定义	133
6.5.2	基于关联度分析的测试用例约简	134
6.5.3	功能模块关联分析	134
6.6	应用与分析	136
6.6.1	覆盖率度量方法应用实例	136
6.6.2	关联度分析方法应用实例	137
	小结	139
第7章	自动化测试工具与应用	140
7.1	软件自动化测试概述	140
7.2	自动化测试的原理和方法	141
7.3	自动化测试的适用范围	143
7.4	避免自动化测试误区	144
7.5	自动化测试工具	146

7.5.1	AutoTestTool	146
7.5.2	Code Analyzer	151
7.5.3	PerformanceRunner	159
7.5.4	AutoRunner	178
7.5.5	TestCenter	194
小结.....		215
附录 A 差异文档(UOP 2.0 与 OOXML 部分功能点的差异)		216
附录 B 功能点转换范围文档(PPT Scope 文档)		224
附录 C 测试用例设计规范文档(UOP 2.0—>OOXML)		233
参考文献.....		244
结束语.....		246

第1章 绪论

1.1 背景介绍

文档是信息的载体。随着多媒体和人机交互技术的发展,文档不仅是纸面介质上的内容,也扩展到声、图、文、像等多媒体数据。电子文档就是以数据方式存于计算机中的文件,计算机是处理电子文档的工具。

随着无纸化办公时代的到来,办公软件已经成为人们日常办公、学习所必需的应用软件,日常办公常用的软件主要包括文字处理(Word)、电子表格(Spreadsheet)以及演示文稿(Presentation)等,都广泛应用于政府、企业、学校和家庭场所。专著中涉及的文档主要是指用办公软件编辑产生的 Word、Excel、PPT 等电子文档。

文档是使用最普遍的信息资源,文档格式标准是目前信息领域最基础、应用最广泛的标准之一。文档格式描述了电子文档的信息编码和存储方式。早期大部分的办公文档格式(例如 Microsoft Office 97~Office 2003)是以二进制的形式存储的,其表示形式是由办公软件厂商所定义的,与其创建程序结合紧密^[1]。封闭的文档格式和算法会给文档的安全带来长期的巨大隐患:首先,用户无法判断文档保存的信息与自己真正保存的内容是否一致,在办公软件版本更替时可能会产生文档信息的失真。其次,文档的兼容能力差,文档格式的私有性导致其他软件对文档无法全部准确解析。再次,文档的生命周期受到格式指定组织及软件开发商的制约,一旦这个组织不再支持此格式,文档信息将面临无法解析的困境^[2]。如果这些办公文档以某个公司私有的文档格式保存,那就意味着政府所有的公文都必须由该公司提供的软件才能使用。如果仅依赖一家厂商的办公软件产品,不能保证政府公文获得持久的支持,这是很不合理的。所以,办公文档广泛的应用不仅要求一种办公文档格式能够被不同办公软件理解和应用,而且也要求不同的文档格式之间能够相互转换。因此,文档格式走向开放成为办公软件用户的共识。

XML(eXtensible Markup Language)^[3]文档以纯文本格式记录,文档中的标记可扩展、易于理解,是业界广为认同的置标语言标准。随着 XML 文档格式的广泛使用,办公文档通过统一制定的、XML 描述的文档格式标准实现了在不同办公软件之间的互通。

OASIS 标准组织制定了一种开放文档格式(Open Document Format,ODF)^[4],于 2006 年被接纳为国际标准,编号为 ISO/IEC 26300。2005 年,微软公司摒弃了封闭文档格式,制定了 OOXML(Office Open XML Format)标准^[5],并于 2008 年 3 月获得批准成为国际标准,编号为 ISO/IEC 29500;它是基于 XML 描述和开放打包规范而定义的一种电子文档格

式标准,该标准定义电子文档是个由多个部件组成的容器,每个部件描述了文档的不同部分,除了少数代表图形的二进制文件或以内嵌形式存在的 OLE 对象,大多数文档部件是基于 XML 格式的^[6]。2002 年,我国开始了中文办公软件文档格式规范(Uniform Office Format, UOF)^[7]的制定,并于 2007 年正式作为国家标准进行推广,编号为 GB/T 20916—2007。

文档格式标准制定的目标在于统一文档格式,然而办公文档格式形成了多个标准共存、标准间互不兼容的现状。但试图以单一的格式标准解决文档之间互操作的问题是不现实的,因为各种文档格式标准具有不同的开发目的、发展背景和用户习惯,短时间内不可能实现多种标准的融合。所以,在多种标准共存的情况下,通过文档格式转换实现标准之间的互操作是必要的^[8]。

ODF/OOXML 转换器,UOF/ODF 转换器和 UOF/OOXML 转换器现均已发布,基本上实现了以 XML 描述的开放文档格式的互通。其中,UOF/OOXML 转换器是微软与北京航空航天大学、北京信息科技大学等合作开发的开源项目,用于实现办公文档在中国国家标准 UOF 和国际标准 OOXML 之间的双向转换,保证文档信息的共享以及互操作^[9]。目前项目已经完成了 6 期的研发任务,正在进行第 7 期的转换器开发和测试,专著的研究内容来源于此项目的测试工作。

文档格式转换项目测试工作中存在很多问题,例如,办公文档的功能点数量繁多、种类复杂,测试人员需要做大量重复的测试用例设计与文档对比工作,测试效率低下;人工设计的测试用例覆盖率无法度量,难以保证测试的完备性;集成测试用例的一级功能点平行组合的方式不够严谨,覆盖率也很低;利用办公软件界面测试转换器的方案存在一定的测试盲点。为解决人工测试工作中存在的问题,专著以文档格式标准 OOXML 和 UOF 为例,探究以 XML 描述的文档格式转换测试的自动化实现。

随着我国“标文通”标准的制定,国产办公软件厂商能有效掌握国内市场竞争的主动权,削弱 20 世纪 90 年代以来微软 Office 垄断市场的局面,摆脱过去高成本兼容微软 Office 文档格式的被动事实,获得更大的利益和生存空间。同时通过制定“标文通”标准,国产办公软件厂商可以利用技术壁垒这种新的方式来保护整个产业的利益。而支持 OOXML 格式标准的微软办公软件的用户群是庞大的。因而实现我国文档格式标准 UOF 与国际标准 OOXML 之间的信息互通意义重大。同时,为了保证办公软件之间的兼容性,实现不同格式标准之间的文档转换是形势所趋,文档格式转换的测试是保证转换质量的重要手段。此专著的研究意义在于,研究文档格式转换的自动化测试,解决人工测试的问题,并为 XML 描述的文档之间转换的测试提供方法参考。

1.2 文档格式转换研究现状

1.2.1 主流文档格式转换器

文档格式转换是实现多种文档格式间互操作的一种有效方法。近年出现了多种格式转换的方法,其中一种就是通过文档格式转换器实现的,它是基于格式规范来设计和实现的,例如,UOF/ODF 转换器、ODF/OOXML 转换器和 UOF/OOXML 转换器,通过转换基本实现了三种文档格式标准 UOF、ODF 和 OOXML 之间的互操作。另外还可以通过办公软件

实现格式转换,例如,红旗 2000 的 RedOffice 和中标普华 Office 均把 ODF 作为默认格式,这些厂商又作为 UOF 标准工作组的成员,对 UOF 提供了很好的支持,通过这些产品也可以很好地实现 ODF 和 UOF 之间的转换。转换器的研制给文档格式间的互操作提供了很大的帮助。目前,UOF、ODF 和 OOXML 三种主流文档格式已经借助转换器实现了互通。

ODF/OOXML 转换器项目是由微软与 Novell 等合作开发的开源项目,包括 2007 年 2 月发布的 ODF Add-in-for-Word、2007 年 5 月发布的 ODF Add-in for PowerPoint 和 ODF Add-in for Excel 成果形式为 Microsoft Office 插件,支持 ODF 和 OOXML 之间的转换。从 2007 年开始,根据用户的反馈与需求,ODF/OOXML 转换器相继发布了 1.0 版本、1.1 版本、2.0 版本、2.5 版本、3.0 版本,到 2010 年 6 月 18 日 ODF/OOXML 4.0 版本正式发布,该版本采用了 OpenXML 的 ISO 新标准(ISO 29500),对 3.0 版本的功能点进行了更新。ODF/OOXML 转换器采用了 .NET Framework 和 XSLT 技术组合的方式实现文档格式的转换。

UOF/ODF 转换器是第一个开放办公文档格式之间的转换器,是由 IBM 与北京大学合作开发的开源项目,2005 年 11 月启动,2006 年 11 月 1.0 版本完成并发布,用于实现国家标准 UOF 与国际标准 ODF 文档格式的双向转换。此转换器主要采用 Java 语言中对 XML 文档按 SAX 方式进行处理,通过多轮 SAX 解析实现文档格式的转换。

UOF Translator 是微软与北京航空航天大学、北京信息科技大学等合作开发的开源项目,用于 UOF 与 OOXML 文档格式的转换,于 2007 年 4 月发起,2008 年 1 月 UOF Translator 1.0 版本正式发布,用于 UOF 与 OOXML 字处理文档格式的转换;2009 年 6 月 5 日,UOF Translator 2.0 版本正式发布,UOF Translator 2.0 实现了 OpenXML (ECMA, 2006. 11)和 UOF 1.0 格式文档之间的双向转换,包括 Word、Spreadsheet、Presentation 三个部分。2010 年 5 月,UOF Translator 2.1 版本发布,UOF Translator 2.1 对上期遗留问题做了修改更新,在电子表格和演示文稿部分增加了新的功能点的转换,并且文字处理部分已经实现了 UOF 1.1 和 OpenXML (ECMA, 2006. 11)格式文档之间的双向转换。2011—2013 年发布了 UOF Translator 3.0、UOF Translator 4.0 和 UOF Translator 4.1,基本上实现了 UOF 与 OOXML 文档格式之间的转换,包括 Word、Spreadsheet、Presentation 三部分。

1.2.2 文档格式转换开发技术研究现状

目前国内外为实现文档格式转换对 XML 文档处理的技术包括以下几种。

1. 利用 DOM 解析器实现

DOM(Document Object Model)是一个与平台、语言无关的程序接口。在解析 XML 文档时,DOM 解析器将整个 XML 文档装入内存,在内存中生成一棵 XML 文档结构树,文档树中的每个节点代表 XML 文档中的一个对象。DOM 解析器还提供了动态访问和更新 XML 文档中的各个节点内容的接口,开发人员可以使用这些接口操纵文档树,在解析文档的时候可以遍历树的任意节点,也可以通过添加、删除或更新树中的节点把源文档转换为其他结构的文档。

DOM 解析的优势在于能够灵活地处理文档,对文档的内容和结构进行直接修改,给转

换带来了很大的便利,且由于 DOM 将整个文档树放入内存,更适合对文档进行随机存取,在转换的时候能够很容易地得到文档信息。然而,当源文档比较大或只需要处理源文档的一小部分时,采用 DOM 就会占用很大的内存资源,低效且浪费资源。

2. 利用 SAX 解析器实现

SAX(Simple API for XML)也是一组得到了广泛使用的程序接口,但其解析器并不像 DOM 解析器那样在内存中建立一个完整的文档树,而是将 XML 文档视为文字流,在解析 XML 文档时根据所遇到的 XML 标签向事件处理程序发出一系列事件,如一个元素的开始、结束、内容等,开发人员可以通过在事件处理程序中编写代码修改这些事件,进而可以实现对源文档的更新和转换。

SAX 最大的优点是内存消耗小,由于整个文档无须一次加载到内存中,这使 SAX 解析器可以解析大于系统内存的文档,但是访问当前节点的同时无法访问前面已经访问过的节点,在格式转换的过程中会增加转换的难度,给格式转换带来一定的局限性,灵活性低。

3. 利用 XSLT 式样单转换

XSLT(eXtensible Stylesheet Language Transformation,可扩展样式单语言转换)可将源 XML 文档的内容转换为另一种格式或结构不同的文档。XSLT 定义了一组转换规则,实现从一个输入产生一个输出。一个 XSLT 式样单由一系列模板组成,模板包含一个表达式,用于描述某些 XML 片段的形式,还包含一个或多个元素,如果这个形式和源文档的内容匹配,那么这些元素就会被输出。这些表达式采用 XPath 技术帮助 XSLT 在 XML 源文档中查找定位信息,通过路径定位文档中需要表示的节点。XPath 本质上是与具有层次结构的 XML 数据模型相匹配的查询语言,可以匹配 XML 文档的任何元素与属性,还包括用于基本字符串处理、数字计算和布尔运算的工具。XSLT 同过程语言不同,它侧重于数据而非代码主体,它在数据内移动,直到找到需要的内容,并且在找到后根据需要进行相关的处理。

在文档格式转换的过程中,可以考虑分层实现某项转换,在主转换之前加入预处理进行必要的结构调整,或增加必要的后处理以得到符合要求的数据内容。预处理和后处理可以采用 DOM 或 SAX 技术实现,主转换采用 XSLT 语言结合 XPath 简捷高效地实现 XML 文档之间的转换。ODF/OOXML、UOF/OOXML 转换器就是采用预处理、主转换、后处理相结合的格式转换方案。另外,ODF/UOF 转换器是采用多轮 SAX 解析方式进行处理,根据元素对应模式的不同,设计出几种相应的转换算法,大大减小了转换难度。

1.2.3 文档格式转换测试技术研究现状

目前针对文档格式转换的测试主要是人工测试,测试人员采用黑盒测试方法测试转换器在功能、性能、系统等方面是否正常工作。测试工作中最耗费人力的工作体现在测试用例设计与文档对比工作上。这是因为文档格式转换项目的测试用例比较特殊,是以办公文档为载体,编辑操作办公软件中的一些文档格式按钮,依据其在办公软件中的显示效果进行设计而来的办公文档。测试执行主要是在办公软件中对比文档格式在转换前后的差异。

针对文档格式转换的自动化测试也有一些初步探索。有人提出用 QTP 等自动化测试

工具,录制对办公软件界面上按钮的操作,生成测试脚本,参数化测试输入,通过软件界面的捕获回放来生成测试用例。但是这种测试方式使用商业化测试工具 QTP,成本高昂,同时增加了额外的测试脚本维护工作。中国电子技术标准化研究所有人提出使用 XMLSpy 等 XML 编辑器编写 XML 代码生成测试用例文档,但是这种方法效率低,要求测试人员具有较高的专业水平。由于生成的 XML 实例文档需要符合相应 XML 模式的各种具体要求,所以生成正确的实例文档并不容易,需要有工具的支持。武汉理工大学提出了一种基于 Form 表单的方法,该方法能根据 XSD 模式文档自动生成 Form 表单,待用户在 Form 表单中输入数据后,自动生成有效的 XML 实例文档。但是办公文档功能点种类繁多,针对每个功能点设计 Form 表单是比较浪费时间的。

1.3 技术分析

为了对两种文档格式标准转换效果进行测试,人工测试是通过支持标准的办公软件的视图化界面对功能点设计用例,输入转换器转换,借助软件视图界面人工对比转换前后两种文档的功能点属性。专著中的技术特点是不依赖办公软件的界面显示,通过分析文档底层的 XML 代码,研究文档格式转换的自动化测试方法。

ZIP 文件格式是一种文档储存格式,也是一种数据压缩格式(可以单独使用打包功能而不使用压缩功能)。ZIP 通常使用后缀名“.zip”。

OOXML 标准将文档看作一个容器,规定了其中的各个组成部件,所有的部件通过 ZIP 进行打包,除了少数代表图形的二进制数据或内嵌 OLE 对象外,大多数部件都采用 XML 文件形式。UOF 以中文办公软件需求为出发点,采用了 W3C XML Schema 作为描述语言,定义了基于 XML 的中文办公文档格式规范,也采用了 ZIP 压缩打包算法。不论何种格式的测试用例文档,其底层代码都是 XML。以 UOF 字处理模块的测试用例文档为例,其底层核心内容的 XML 代码如图 1.1 所示。

```
<?xml:文字处理文档_4225 xmlns:字="http://schemas.uof.org/cn/2009/wordproc" xmlns:uof="http://schemas.uof.org/cn/2009/uof">
  <字:分节_416A>
    <字:节属性_421B>
      <字:节类型_41EA>continuous</字:节类型_41EA>
    </字:节属性_421B>
  </字:分节_416A>
  <字:段落_416B>
    <字:段落属性_419B 样式引用_419C="id000000">
      <字:句_419D>
        <字:句属性_4158>
          <字:是否粗体_4130>true</字:是否粗体_4130>
        </字:句属性_4158>
        <字:文本串_415B>粗体</字:文本串_415B>
      </字:句_419D>
    </字:段落_416B>
  </字:文字处理文档_4225>
```

图 1.1 测试用例文档底层 XML 代码实例

专著的研究对象是 UOF 和 OOXML 格式的文档。虽然它们是不同的办公文档,但在内容结构上分析两种文档标准,可以发现,UOF 标准和 OOXML 标准在整体结构、功

能点对应以及 XML 描述上存在一致性,可以通过读取这两种文档底层的 XML 代码实现文档格式转换的自动化测试。这也体现了专著在文档格式转换自动化测试方法研究的贡献与创新:

(1) 由于办公文档中的功能点太多,前期测试中人工对比增大了测试员的工作量,专著的研究绕过了前台办公软件的显示界面,通过后台的 XML 代码的提取和对比,直接对转换前后的文档进行对比,降低了测试员的劳动强度,并且可以对大量的文档信息进行测试。

(2) 由于现有软件不能较好地支持 UOF 标准,导致了人工测试中存在测试盲点,专著提出的测试方法避免了由于办公软件不能很好地支持标准所带来的手工测试的错误。

(3) 功能点的明确划分是进行测试用例自动生成和功能点覆盖度量的基础。专著通过对文档格式标准的 XML Schema 定义和实例文档 XML 结构进行形式化描述,定义了独立功能点切片和独立功能点簇的概念,统一了测试功能点划分粒度,以实现办公文档功能点的量化,也为测试用例自动生成提供了依据。

1.4 专著内容

软件测试的目的是设计并执行测试用例(包括测试用例的输入、测试用例执行步骤和执行环境以及测试用例的预期输出)得到实际输出,对比预期输出和实际输出判断测试用例是否通过,以尽早发现程序中所存在的缺陷。对于文档格式转换来说,测试用例的输入是将被转换的测试用例文档,实际输出是经转换器转换后得到的文档,预期输出与实际输出一致是指转换前后文档相应功能点的属性在差异文档允许的范围内保持一致。

办公文档都有基本的构成要素,如元数据、链接与标签、段落、表格、图表等,称之为功能,不同的功能还可具体细分,在办公软件里主要是通过对按钮操作设置的文档具体格式来体现的,比如字体的粗体、段落设置、图形的填充等。专著研究的功能点就是基于办公文档的基本构成要素,测试用例文档是这些功能点的载体。

专著的内容是研究文档格式转换自动化测试,以解决人工测试工作中存在的测试效率低、覆盖率难以度量等问题,主要包括测试用例自动生成、功能点自动对比和自动化测试过程管理。此外,也会相应地介绍一些经典的测试方法以及自动化测试工具。

测试用例自动生成主要是以现有的 UOF 2.0 文档格式标准和转换器需求规格说明相关文档为依据,生成满足 UOF 2.0 格式标准的测试用例。以电子表格模块为例,主要的研究内容是通过逻辑层 XML Schema 格式的文档标准进行分析,根据其数据特点建立概念层的 UML 类模型,设计基于 .NET 平台的面向对象的算法,实现对物理层的电子表格文档 XML 进行操作,以自动生成测试用例。具体包括以下几点。

(1) 根据对测试需求的形式化定义,划分测试功能点,提供覆盖 UOF 2.0 标准的测试功能点列表,实现测试用例覆盖率的度量,降低测试遗漏和冗余的风险。研究组合覆盖法生成集成测试数据的方案,用更简洁的测试用例覆盖更多的测试功能点组合。

(2) 根据测试设计人员选择的测试功能点和输入的测试数据,自动生成相应的测试用例(转换器源文档),该测试用例能够通过 UOF 2.0 标准 XML Schema 的格式来验证。

(3) 实现参数化的数据驱动测试用例自动生成。测试数据存储在外部文件或数据库中,与测试用例文档分离,可以方便地修改或重用。在系统运行时动态地读取测试数据,生

成测试用例文档,能快速高效地实现多次迭代。

(4) 测试用例的设计脱离对办公软件界面的依赖,避免办公软件对标准的支持不够所产生的测试盲点问题。

用功能点在转换前后式样和内容的对比来验证转换器功能的正确性。人工对比是在办公软件里比对各个功能点式样与内容,专著研究的功能点自动对比脱离对办公软件的依赖,减少人工操作办公文档的工作量,避免人工对比存在的主观偏差,实现功能点的自动对比。具体研究内容为以下几点。

(1) 提取源文档(UOF 2.0 文档)与目标文档(OOXXML 文档)中待测试的功能点 XML 信息。

(2) 建立 UOF 和 OOXXML 格式功能点属性的预处理器,将第一个步骤提取出来的源文档和目标文档待测功能点属性,分别由 UOF 预处理器和 OOXXML 预处理进行属性的预处理,得到源文档和目标文档的功能点属性格式化数据,即最终属性,将得到的两种标准的最终属性进行存储。

(3) 建立信息比较器,然后进行最终属性数据的分析比较。

自动化测试过程管理主要包括两个方面:测试用例管理和缺陷管理。

(1) 建立测试用例管理模型,通过覆盖率度量自动判定生成的测试用例是否满足覆盖率要求,满足要求的用例自动执行测试。

(2) 以 UOF 2.0 文档格式标准和转换器需求规格说明等相关文档为依据,从测试用例底层 XML 代码入手,分析 UOF 2.0 Schema 文档数据特点,测试功能点之间的关联关系,设计覆盖率度量方法和关联度分析方法。

(3) 建立缺陷管理模型,把自动对比分析过程中发现的问题自动提交到缺陷管理系统中,自动生成缺陷表单,并实现对修改后的代码自动进行验证。

第2章 文档格式转换测试

目前,办公文档有多个标准共存,不同的标准有不同的开发目的、发展背景等,并且不同标准在内容结构、计量单位以及描述方式上有很大差异,在短期内不可能实现多个标准的融合,通过文档转换器实现文档信息在标准间的共享和互操作是必要的。文档转换器的测试对指导转换器的开发,评测开发的质量有着重要意义。

本章以项目组的 UOF/OOXML 转换器为例,介绍转换器的整个测试过程及测试的相关理论。该转换器为了实现办公文档在 OOXML 与 UOF 标准之间的转换,测试工作采用了黑盒测试的思想,按照测试需求中转换器实现的功能点来设计测试用例,然后输入转换器转换,对比转换前后的文档内容和格式是否在差异文档(见附录 A)允许范围内。

下面以工作流程为主线,以项目组第 6 期中演示文稿部分测试工作为例来介绍项目的整个测试过程。

2.1 文档格式转换测试过程概述

项目组从 2007 年至今已完成了 6 期的开发和测试工作,分别实现了 UOF 1.0、UOF 1.1、UOF 2.0 到 ECMA 376、ISO/IEC 29500 之间的双向转换,内容包括 Word、Spreadsheet、Presentation 三部分。前期的测试主要是人工测试,项目组提出借助支持不同文档标准的图形化界面办公软件设计测试用例,执行文档转换器转换,并利用办公软件人工对比转换前后的测试用例是否通过。UOF/OOXML 文档转换器的人工测试步骤如图 2.1 所示。

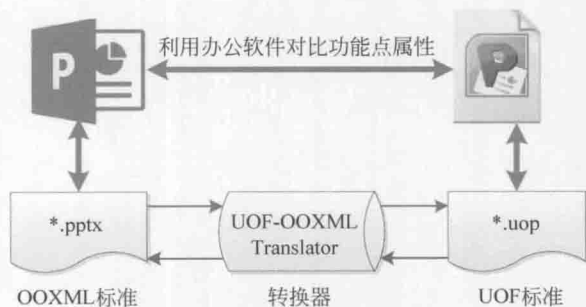


图 2.1 文档转换器测试的整体思想

首先由测试设计人员通过分析产品需求中所列出的转换范围,设计基于单个功能点的和基于功能点集成的测试用例(源文档);然后由测试执行人员将测试用例通过 UOF/