



(12)发明专利申请

(10)申请公布号 CN 110888808 A

(43)申请公布日 2020.03.17

(21)申请号 201911122743.5

(22)申请日 2019.11.16

(71)申请人 云南湾谷科技有限公司

地址 650051 云南省昆明市盘龙区东华街
道办事处小龙村益鑫德隆小区6幢17
层1705室

(72)发明人 周龙

(51)Int.Cl.

G06F 11/36(2006.01)

G06F 16/36(2019.01)

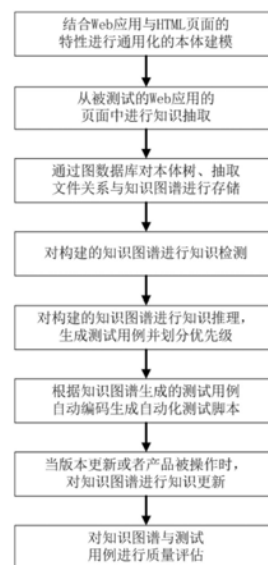
权利要求书2页 说明书7页 附图6页

(54)发明名称

一种基于知识图谱的Web智能化测试的方法

(57)摘要

本发明公开了一种基于知识图谱的Web智能化测试的方法,包括:步骤一,结合Web应用与HTML页面的特性进行通用化的本体建模;步骤二,从被测试的Web应用的页面中进行知识抽取;步骤三,通过图数据库对本体树、抽取文件关系以及知识图谱进行存储;步骤四,对知识图谱进行知识检测;步骤五,对知识图谱进行知识推理,生成测试用例;步骤六,根据测试用例编写自动化测试脚本,执行测试并生成测试报告;步骤七,当版本迭代或者进行操作时,对知识图谱进行知识更新;步骤八,对知识图谱进行质量评估。本发明弥补了知识图谱在测试领域还没有应用的空白,实现了无需人工参与的情况下完成测试流程的设计、维护、自动化执行、自我优化与持续更新。



1. 一种基于知识图谱的Web智能化测试的方法,其特征在于,包括:步骤一,结合Web应用与HTML页面的特性进行通用化的本体建模;步骤二,从被测试的Web应用的页面中进行知识抽取;步骤三,通过图数据库对本体树、抽取文件关系以及知识图谱进行存储;步骤四,对知识图谱进行知识检测;步骤五,对知识图谱进行知识推理,生成测试用例;步骤六,根据测试用例编写自动化测试脚本,执行测试并生成测试报告;步骤七,当版本迭代或者进行操作时,对知识图谱进行知识更新;步骤八,对知识图谱进行质量评估。

2. 根据权利要求1所述的基于知识图谱的Web智能化测试的方法,其特征在于,所述的步骤一中,本体建模结合HTML页面与Web应用的特性采用自顶向下方式进行构建;包括:

步骤S11,结合Web应用与HTML页面的特性采用自顶向下的方式对实体类、实体类层级进行构建:根据HTML页面的特性,将HTML标签作为本体建模的基础,根据HTML的分类、特性、内容以及样式进行展开,形成基础实体类以及实体类层级,再根据Web应用的特性,将Web应用设计中模块、功能等抽象化概念作为实体类以及实体类层级的补充,结合形成完整的实体类以及实体类层级;其中,实体类与实体类层级采用树结构进行表示;

步骤S12,结合Web应用与HTML页面的特性采用自顶向下的方式对关系类、关系类层级进行构建:根据关系是固有属性还是可执行操作划分为动态关系类和静态关系类两个基础关系类,在此继续上根据固有属性的特征与可执行操作的类型在基础关系类下继续细化分支,形成关系类层级;其中,关系类与关系类层级采用树结构进行表示;

步骤S13,通过对HTML、CSS规范中标签的特性进行通用化与抽象化的构建,形成对实体类的固有的、精确的、可表达的描述,作为实体类属性;

步骤S14,通过对HTML、CSS、JS规范中层级、样式、条件以及事件进行归类与抽象,形成对关系类的固有的、精确的、可唯一表达的或者可表示执行过程的描述,作为关系类属性。

3. 根据权利要求1所述的基于知识图谱的Web智能化测试的方法,其特征在于,所述步骤二中,知识抽取遍历从实际运行的Web应用中爬取的HTML页面文件、JS 文件、CSS 文件,通过对文件关联关系与文件内容进行解析,抽取实体、实体属性、关系以及关系属性;包括:

步骤S21,爬取的HTML页面文件、JS文件、CSS文件的过程中,记录入口页面;爬取完成后将文件目录与代码库中的文件目录进行比对,确保爬取文件的完整性;

步骤S22,从入口页面开始抽取,通过HTML页面的标签关联需要抽取的JS文件和CSS文件,通过HTML文件标签和JS文件函数的跳转关系关联需要抽取的下一个HTML页面;

步骤S23,抽取过程通过抽取关系图进行记录,根据图的遍历算法设计抽取顺序,以确保整个抽取过程不重复、不遗漏;

步骤S24,抽取HTML文件中有唯一标签、类标签、绑定数据以及绑定事件的标签作为实体;抽取HTML文件中标签的层级和对应的响应事件以及JS文件中的事件函数作为关系;抽取HTML文件中标签的属性以及CSS文件中标签的样式的抽象作为实体属性;抽取HTML文件中标签的层级、响应事件条件以及JS文件中事件函数执行的条件、过程和结果作为关系属性。

4. 根据权利要求1所述的基于知识图谱的Web智能化测试的方法,其特征在于,所述步骤三中,通过图数据库对本体树、抽取文件关系以及知识图谱进行存储。

5. 根据权利要求1所述的基于知识图谱的Web智能化测试的方法,其特征在于,所述步

骤四中,知识检测对知识图谱中存在的冲突、缺失、重复以及较复杂的实体、实体属性、关系和关系属性进行检测,包括:

步骤S41,冲突检测对实体属性冲突、实体间关系冲突以及关系属性冲突进行检测;

步骤S42,缺失检测对孤立实体(与其他任何实体均无关联关系的实体)、孤立关系(只与一个实体连接的关系)、缺失实体属性以及缺失关系属性进行检测;

步骤S43,重复检测对重复实体(关系、属性完全相同的实体)、重复关系(关联实体、属性完全相同的关系)、重复实体属性、重复关系属性进行检测;

步骤S44,复杂性检测对复杂实体(属性或关联关系过多)、复杂关系(关联过多实体的关系)、复杂路径(两个实体间最短路径大于整体最短路径2倍的)进行检测。

6. 根据权利要求1所述的基于知识图谱的Web智能化测试的方法,其特征在于,所述步骤五中,对知识图谱进行知识推理,形成测试用例并划分优先级;具体包括:

步骤S51,根据深度优先遍历算法对知识图谱中由动态关系关联的实体进行遍历,遍历的路径作为基础测试用例;

步骤S52,根据路径的长度、路径中所有动态关系的权重、所有节点的入度和出度、所有节点的迭代版本数对基础测试用例进行加权,根据加权的结果划分测试用例的优先级。

7. 根据权利要求1所述的基于知识图谱的Web智能化测试的方法,其特征在

于,所述步骤六中,根据要求6所述的测试用例编写自动化测试脚本。

8. 根据权利要求7所述的自动化测试脚本,其特征在于,自动化测试脚本按照基础模块与执行模块进行划分,其中基础模块对页面、页面功能、页面控件操作进行封装,具体地,页面封装为包、页面功能封装为文件中的类、页面操作封装为类方法;执行模块按照测试用例的路径对应地调用基础模块的类方法。

9. 根据权利要求1所述的基于知识图谱的Web智能化测试的方法,其特征在于,所述步骤七中,通过知识融合和属性更新进行知识更新;具体包括:

步骤S71,当Web应用版本更新迭代时,对新版本进行权利要求3所述知识抽取形成新的知识图谱,通过知识融合将新的实体、实体属性、关系以及关系属性与构建好的知识图谱进行整合形成新的完整的知识图谱;

步骤S72,当Web应用被操作时,包括包括用户操作与自动化测试脚本操作,对知识图谱中的实体属性与关系属性进行更新。

10. 根据权利要求1所述的基于知识图谱的Web智能化测试的方法,其特征在于,所述步骤八中,质量评估涵盖对知识图谱中实体、关系、属性的涵盖率、准确率、有效率、一致率进行评估以及测试用例的涵盖率、准确性、有效性进行评估。

一种基于知识图谱的Web智能化测试的方法

[0001]

技术领域

[0002] 本发明涉及软件测试领域以及人工智能领域,特别是一种基于知识图谱的Web智能化测试的方法。

[0003]

背景技术

[0004] 随着信息技术的飞速发展,Web应用因其易用性、通用性和高扩展性逐渐成为当下的主流应用,涉及到了政务、信息化、社交、购物、娱乐、管理等多个层面。同时,随着市场环境的快速变化,用户需求的不断增长,Web应用的迭代更新也变得越来越来频繁,由此,给开发测试提出了更高的要求,也使得测试工作变得越来越复杂、工作量越来越大、响应速度也越来越快。

[0005] 为了应对长期的、复杂的和大工作量的测试工作,自动化测试作为手工测试的补充应运而生,通过自动化测试技术,程序可以代替人工进行重复的测试工作,且效率和准确率更高。但自动化测试在减少测试工作量的同时,也带来了巨大的设计、开发和维护工作量,因为当前的自动化测试首先需要对测试过程进行精细的设计,再根据过程设计编写繁杂的自动化测试脚本,同时,随着软件需求功能的变更,测试脚本也需要随之修改,这就导致自动化测试脚本的设计、组织、开发和维护变得极其复杂和不稳定,需要消耗大量的开发资源,且重用性、通用性非常的差。使得在多数项目中,自动化测试的成本甚至比手工测试要大得多。

[0006] 随着大数据与人工智能技术的发展,各个领域都在逐渐实现自动化向智能化的转化,使得机器不再只完成重复的动作,而且像人类一样学习“知识”、积累“经验”,并根据这些“知识”和“经验”进行识别、推理以及决策。而知识图谱作为重要的人工智能技术,被广泛的应用于智能搜索、过程优化以及信息推理等方向。其中,知识图谱强大的知识构建和知识推理能力也为智能化测试提供了重要基础,通过对被测试系统进行本体建模、知识构建,可以形成被测试系统的完整架构、功能体系,在此基础上,通过知识推理智能生成可靠、动态的测试用例和测试脚本,

并自动化执行,同时,当功能需求修改时,通过知识融合不断更新知识图谱。与当前的大量借助人工进行设计、开发和维护的自动化测试相比,基于知识图谱的Web智能化测试能够在极少量人工参与的情况下完成测试的设计、维护、执行与自我优化等工作,同时也弥补了知识图谱作为全新的技术,在测试领域还没有应用的空白。

[0007]

发明内容

[0008] 本发明的目的在于提供一种基于知识图谱的Web智能化测试的方法,有效构建智能化测试,降低自动化测试工作设计、开发和维护的工作量。

[0009] 实现上述目的的技术方案是：一种基于知识图谱的Web智能化测试，包括：

步骤一，结合Web应用与HTML页面的特性进行通用化的本体建模；步骤二，从被测试的Web应用的页面中进行知识抽取；步骤三，通过图数据库对本体树、抽取文件关系以及知识图谱进行存储；步骤四，对知识图谱进行知识检测；步骤五，对知识图谱进行知识推理，生成测试用例；步骤六，根据测试用例编写自动化测试脚本，执行测试并生成测试报告；步骤七，当版本迭代或者进行操作时，对知识图谱进行知识更新；步骤八，对知识图谱进行质量评估。优选的，所述的步骤一中，本体建模结合HTML页面与Web应用的特性采用自

顶向下方式进行构建，包括：

步骤S11，结合Web应用与HTML页面的特性采用自顶向下的方式对实体类、实体类层级进行构建。其中，根据HTML页面的特性，将HTML标签作为本体建模的基础，再按照HTML标签的分类、特性、内容以及样式进行展开，形成基础实体类以及实体类层级，在此基础上，通过Web应用设计中模块、功能等抽象化概念进行补充，形成完整的实体类以及实体类层级。其中，实体类与实体类层级采用树结构进行表示，从属关系的实体类在树结构中表示为父节点与子节点，同级关系的实体类在树结构中表示为同一层节点；

步骤S12，结合Web应用与HTML页面的特性采用自顶向下的方式对关系类、关系类层级进行构建：根据关系是固有属性还是可执行操作划分为动态关系类和静

态关系类两个基础关系类，在此继续上根据固有属性的特征与可执行操作的类型在基础关系类下继续细化分支，形成关系类层级。其中，关系类与关系类层级采用树结构进行表示，从属的关系在树结构中表示为父节点与子节点，同级的关系在树结构中表示为同一层节点；

步骤S13，通过对HTML、CSS规范中标签的特性进行通用化与抽象化的构建，形成对实体类的固有的、精确的、可表达的描述，作为实体类属性；

步骤S14，通过对HTML、CSS、JS规范中层级、样式、条件以及事件进行归类与抽象，形成对关系类的固有的、精确的、可唯一表达的或者可表示执行过程的描述，作为关系类属性；

优选的，所述步骤二中，知识抽取遍历从实际运行的Web应用中爬取的HTML页面文件、JS文件、CSS文件，通过对文件关联关系与文件内容进行解析，抽取实体、实体属性、关系以及关系属性；包括：

步骤S21，爬取的HTML页面文件、JS文件、CSS文件的过程中，记录入口页面；爬取完成后将文件目录与代码库中的文件目录进行比对，确保爬取文件的完整性；

步骤S22，从入口页面开始抽取，通过HTML页面的标签关联需要抽取的JS文件和CSS文件，通过HTML文件标签和JS文件函数的跳转关系关联需要抽取的下一个HTML页面；

步骤S23，抽取过程通过抽取关系图进行记录，根据图的遍历算法设计抽取顺序，以确保整个抽取过程不重复、不遗漏；

步骤S24，抽取HTML文件中有唯一标签、类标签、绑定数据以及绑定事件的标签作为实体；抽取HTML文件中标签的层级和对应的响应事件以及JS文件中的事件函数作为关系；抽取HTML文件中标签的属性以及CSS文件中标签的样式的抽象作为实体属性；抽取HTML文件中标签的层级、响应事件条件以及JS文件中事件函数执行的条件、过程和结果作为关系属性；

优选的，所述步骤三中，通过图数据库对本体树、抽取文件关系以及知识图谱进行存

储;

其中,本体树中实体分类与关系分类将对通过所属步骤二中知识抽取构建的知识图谱中的实体与关系进行精确的定位与层级的关联;抽取文件关系存储在图数据库

库中,通过图遍历算法决定抽取的顺序与关联;优选地,针对知识图谱的存储,实体存储为图数据库中的节点,关系存储为图

数据库中节点间的连线,实体属性存储为图数据库中节点的属性,关系属性存储为图数据库中连线的属性;

优选的,所述步骤四中,知识检测指:对知识图谱中存在的冲突、缺失、重复以及较复杂的实体、实体属性、关系和关系属性进行检测,以提升知识图谱与通过知识图谱推理的测试用例的有效性,同时对整个 Web平台产品设计提出优化建议;具体包括:

步骤 S41,冲突检测对实体属性冲突、实体间关系冲突以及关系属性冲突进行检测;

步骤 S42,缺失检测对孤立实体(与其他任何实体均无关联关系的实体)、孤立关系(只与一个实体连接的关系)、缺失实体属性以及缺失关系属性进行检测;

步骤 S43,重复检测对重复实体(关系、属性完全相同的实体)、重复关系(关联实体、属性完全相同的关系)、重复实体属性、重复关系属性进行检测;

步骤S44,复杂性检测对复杂实体(属性或关联关系过多)、复杂关系(关联过多实体的关系)、复杂路径(两个实体间最短路径大于整体最短路径2倍的)进行检测;

优选的,所述的步骤五中,对知识图谱进行知识推理,形成测试用例并划分优先级;

其中,根据深度优先遍历算法对知识图谱中由动态关系关联的实体进行遍历,遍历的路径作为基础测试用例。在此基础上,通过路径的长度、路径中所有动态关系的权重、所有节点的入度和出度、所有节点的迭代版本数对基础测试用例进行加权,根据加权的结果划分测试用例的优先级;

优选的,所述的步骤六中,对知识图谱进行知识推理,形成测试用例并划分优先级;根据步骤五所述的测试用例编写自动化测试脚本;

优选的,为了便于自动化测试脚本的维护和二次开发,自动化测试脚本按照基础模块与执行模块进行划分;其中基础模块对页面、页面功能、页面控件操作进行封装,具体地,页面封装为包、页面功能封装为文件中的类、页面操作封装为类方法。执行模块按照测试用例的路径对应地调用基础模块的类方法;

优选的,所述的步骤七中,通过知识融合和属性更新进行知识更新;具体包括:步骤 S71,当Web应用版本更新迭代时,对新版本进行权利要求3所述知识抽

取形成新的知识图谱,通过知识融合将新的实体、实体属性、关系以及关系属性与构建好的知识图谱进行整合形成新的完整的知识图谱;

步骤S72,当Web应用被操作时,包括包括用户操作与自动化测试脚本操作,对知识图谱中的实体属性与关系属性进行更新;

优选的,所述的步骤八中,对知识图谱与生成的测试用例进行质量评估;其中,知识图谱的质量评估包括对实体、关系、属性的涵盖率、准确率、有效

率、一致率进行评估;测试用例的质量评估包括对平台功能的涵盖率、测试结果的准确性以及测试用例的有效性进行评估;

本发明的有益效果是:本发明弥补了知识图谱在测试领域还没有应用的空白,通过知

识抽取构建知识图谱,再对知识图谱进行知识推理,自动编码生成自动化测试脚本,大大提升了 Web应用功能梳理的效率,降低了大量的测试脚本设计、开发和维护的工作量;同时,本发明具有很强的自我更新的能力和适应性,能适应各种规模的Web 应用并与其迭代过程无缝衔接;本发明具备较高的稳定性与可追溯性,较人工设计测试用例在涵盖面、准确性、高效性上有巨大的提升;

附图说明

[0010] 图1为本发明实施例提供的一种基于知识图谱的Web智能化测试的流程图;
图2为本发明实施例提供的一种本体建模中实体类以及实体类层级的示意图;
图3为本发明实施例提供的一种本体建模中关系类以及关系类层级的示意图;
图4为本发明实施例提供的一种实体属性与关系属性的示意图;
图5为本发明实施例提供的一种知识图谱知识抽取的流程图;
图6为本发明实施例提供的一种知识图谱的子图谱的示意图;
图7为本发明实施例提供的一种知识图谱知识检测的流程图;
图8为本发明实施例提供的一种知识图谱知识推理的流程图;
图9为本发明实施例提供的一种知识图谱测试脚本的架构图;
图10为本发明实施例提供的一种知识图谱知识更新的流程图;

具体实施方式

[0011] 下面将结合附图对本发明作进一步说明。

[0012] 请参阅图1,本发明一种基于知识图谱的Web智能化测试,包括:步骤一,本体建模结合HTML页面与Web应用的特性采用自顶向下方式进行构

建。构建内容包括:实体类以及实体层级、关系类以及关系类层级、实体类属性以及关系类属性的构建。具体包括:

步骤S11,结合Web应用与HTML页面的特性采用自顶向下的方式对实体类、实体类层级进行构建:根据HTML页面的特性,将HTML标签作为本体建模的基础,根据HTML的分类、特性、内容以及样式进行展开,形成基础实体类以及实体类层级,再根据Web应用的特性,将Web应用设计中模块、功能等抽象化概念作为实体类以及实体类层级的补充,结合形成完整的实体类以及实体类层级。其中,实体类与实体类层级采用树结构进行表示。如图 2 所示,本体构建中实体类以及实体类层级的示意图,如下:

针对HTML页面的特性,将最小宏观单元“页面”作为实体类的基础类。具体地,在“页面”的基础上进行细化,根据Web应用设计中的功能特性,抽象地将页面中的元素划分为三个子类:“描述”、“呈现”和“交互”;其中,“描述”子类用于表达页面元素的本身特性,“呈现”子类用于归类用于信息展示的元素,“交互”子类用于归类与用户产生界面交互的元素;

具体地,“描述”子类又根据描述元素的不同方面,划分为“内容”和“样式”,“样式”又细分为“颜色”、“高度”等;

具体地,“呈现”子类根据展示的信息的不同,划分为“图片”、“文本”、“动画”等,“文本”类又包含HTML标签中的“<p>”、“”等标签类;

具体地,“交互”子类根据用户界面交互操作的不同,划分为“按钮”、“链接”和“输入框”

等,进一步地,可以结合HTML标准继续进行细化;

步骤S12,结合Web应用与HTML页面的特性采用自顶向下的方式对关系类、关系类层级进行构建:根据关系是固有属性还是可执行操作划分为动态关系类和静态

态关系类两个基础关系类,在此继续上根据固有属性的特征与可执行操作的类型在基础关系类下继续细化分支,形成关系类层级。其中,关系类与关系类层级采用树结构进行表示;如图3所示,本发明构建的关系类以及关系类层级的示意图。其中“关系”基础类划分为“静态”类和“动态”类,“静态”类根据对HTML标签关系的抽象进一步划分为“从属”、“关联”和“等价”等关系子类;“动态”类根据对HTML标签转换形式与用户操作类型的抽象进一步划分为“点击”、“跳转”以及“更新”等关系子类;

步骤S13,通过对HTML、CSS规范中标签的特性进行通用化与抽象化的构建,形成对实体类的固有的、精确的、可表达的描述,作为实体类属性;

步骤S14,通过对HTML、CSS、JS规范中层级、样式、条件以及事件进行归类与抽象,形成对关系类的固有的、精确的、可唯一表达的或者可表示执行过程的描述,作为关系类属性;

如图4所示,本发明的一种实体属性与关系属性的示意图;实体属性通用化与抽象化HTML、CSS规范中标签的特性,形成“分类”、“ID”、“最大长度”、“是否非空”、“是否可编辑”、“语言代码”、“数据类型”等属性;关系属性归类和抽象HTML、CSS、JS规范中层级、样式、条件以及事件,形成“分类”、“ID”、“页面打开方式”、“交互数量”、“版本编号”、“迭代标识”、“移入/移出”等属性。

[0013] 步骤二,从实际运行的Web应用中爬取的HTML页面文件、JS文件、CSS文件作为数据源,通过对这些文件进行知识抽取,抽取实体、实体属性、关系以及关系属性构建智能化测试知识图谱;如图 5 所示,一种知识图谱知识抽取的流程图。步骤S21,爬取的HTML页面文件、JS文件、CSS文件的过程中,记录入口页面;爬取完成后将文件目录与代码库中的文件目录进行比对,确保爬取文件的完整性;

步骤S22,从入口页面开始抽取,通过HTML页面的标签关联需要抽取的JS文件和CSS文件,通过HTML文件标签和JS文件函数的跳转关系关联需要抽取的下一个HTML页面;

步骤S23,抽取过程通过抽取关系图进行记录,根据图的遍历算法设计抽取顺序,直到整个抽取关系图遍历完成,即结束抽取;以确保整个抽取过程不重复、不

遗漏;

步骤S24,抽取HTML文件中有唯一标签、类标签、绑定数据以及绑定事件的标签作为实体;抽取HTML文件中标签的层级和对应的响应事件以及JS文件中的事件函数作为关系;抽取HTML文件中标签的属性以及CSS文件中标签的样式的抽象作为实体属性;抽取HTML文件中标签的层级、响应事件条件以及JS文件中事件函数执行的条件、过程和结果作为关系属性;抽取完成的实体、实体属性、关系、关系属性实时存入图数据库;

步骤三,通过图数据库对本体树、抽取文件关系以及知识图谱进行存储;其中,针对知识图谱的存储,实体存储为图数据库中的节点,关系存储为图数据库中节点间的连线,实体属性存储为图数据库中节点的属性,关系属性存储为图数据库中连线的属性;

经所述步骤二知识抽取,获取智能化测试知识图谱。如图6所示,本发明实施例提供了一种知识图谱的子图谱的示意图。

[0014] 具体地,通过知识抽取,获取了七个实体:“Index”、“Username”、“Password”、

“Submit”、“Info”、“UserCenter”、“密码错误”、“账号不存在”。其中实体的名称即实体的ID,针对页面实体,访问页面的控制器名即为实体名;针对页面元素实体,HTML标签ID即为实体名;针对描述实体,描述的内容或样式的值即为实体名;

具体地,通过知识抽取,获取了四个关系:静态关系“从属”、“关联”和动态关系“点击”、“更新”,其中关系的名称即是关系ID。静态关系表示了实体间固有的关系,比如,“Username”控件属于“Index”页面;动态关系表示了实体间的转换形式,比如,通过点击“Submit”控件可以跳转到“UserCenter”页面;

具体地,通过知识抽取,获取实体属性与关系属性。比如:实体“Username”的部分属性包括:“分类”、“ID”、“是否为空”、“是否可编辑”、“交互数量”、“版本号”,表示“Username”控件是一个可编辑的不能为空的输入框,被输入了10次;关系“更新”的部分属性包括:“分类”、“ID”、“条件”、“交互数量”、“版本号”,表示“Info”当“条件”为“result=101”时将内容更新为“密码错误”;

步骤四,对通过步骤二构建好的知识图谱进行知识检测,并将检测结果生成检测报告,以提升知识图谱与通过知识图谱推理的测试用例的有效性,同时对整个Web平台产品设计提出优化建议。如图7所示,本发明实施例提供的一种知识图谱知识检测的流程图。

[0015] 步骤S41,冲突检测对实体属性冲突、实体间关系冲突以及关系属性冲突进行检测;

步骤 S42,缺失检测对孤立实体(与其他任何实体均无关联关系的实体)、孤立关系(只与一个实体连接的关系)、缺失实体属性以及缺失关系属性进行检测;

步骤S43,重复检测对重复实体(关系、属性完全相同的实体)、重复关系(关联实体、属性完全相同的属性)、重复实体属性、重复关系属性进行检测;

步骤S44,复杂性检测对复杂实体(属性或关联关系过多)、复杂关系(关联过多实体的关系)、复杂路径(两个实体间最短路径大于整体最短路径2倍的)进行检测;

步骤五,对知识图谱进行知识推理,形成测试用例并划分优先级;如图8所示,本发明实施例提供的一种知识图谱知识推理的流程图。具体包括:

步骤S51,根据深度优先遍历算法对知识图谱中由动态关系关联的实体进行遍历,直到遍历完全部实体即结束遍历,遍历的路径作为基础测试用例;

步骤S52,提取生成路径的特征,包括:路径的长度、路径中所有动态关系的权重、所有节点的入度和出度、所有节点的迭代版本数,通过这些特征对路径进行加权,根据加权的結果输出测试用例以及测试用例的优先级;

步骤六,根据步骤五所述的测试用例编写自动化测试脚本;如图9所示,本发明实施例提供的一种知识图谱测试脚本的架构图。自动化测试脚本按照基础模块与执行模块进行划分;其中基础模块对页面、页面功能、页面控件操作进行封装,具体地,页面封装为包、页面功能封装为文件中的类、页面操作封装为类方法。执行模块按照测试用例的路径对应地调用基础模块的类方法生成测试执行文件;

步骤七,通过知识融合和属性更新进行知识更新;如图10所示,本发明实施例提供的一种知识图谱知识更新的流程图。具体包括:

步骤S71,当Web应用版本更新迭代时,对新版本进行权利要求3所述知识抽取形成新的知识图谱,通过知识融合将新的实体、实体属性、关系以及关系属性与

构建好的知识图谱进行整合形成新的完整的知识图谱；

步骤S72,当Web应用被操作时,包括包括用户操作与自动化测试脚本操作,对知识图谱中的实体属性与关系属性进行更新；

步骤八,对知识图谱与生成的测试用例进行质量评估;包括:对知识图谱的实体、关系、属性的涵盖率、准确率、有效率、一致率进行评估;测试用例的质量评估包括对平台功能的涵盖率、测试结果的准确性以及测试用例的有效性进行评估;

以上实施例仅供说明本发明之用,而非对本发明的限制,有关技术领域的技术人员,在不脱离本发明的精神和范围的情况下,还可以作出各种变换或变型,因此所有等同的技术方案也应该属于本发明的范畴,应由各权利要求所限定。

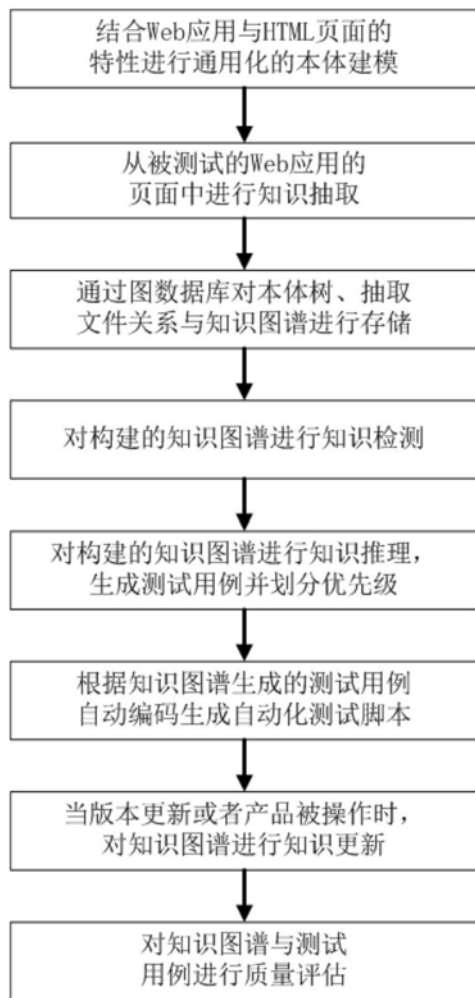


图1

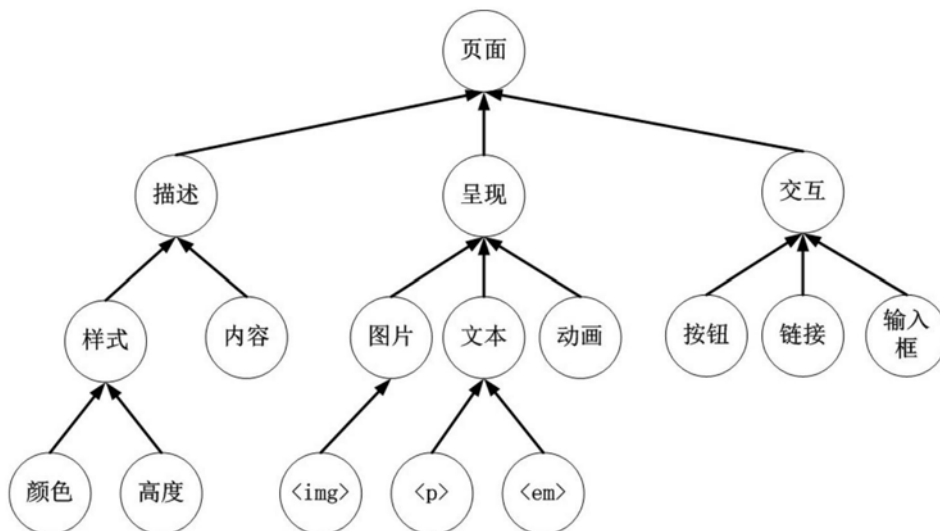


图2

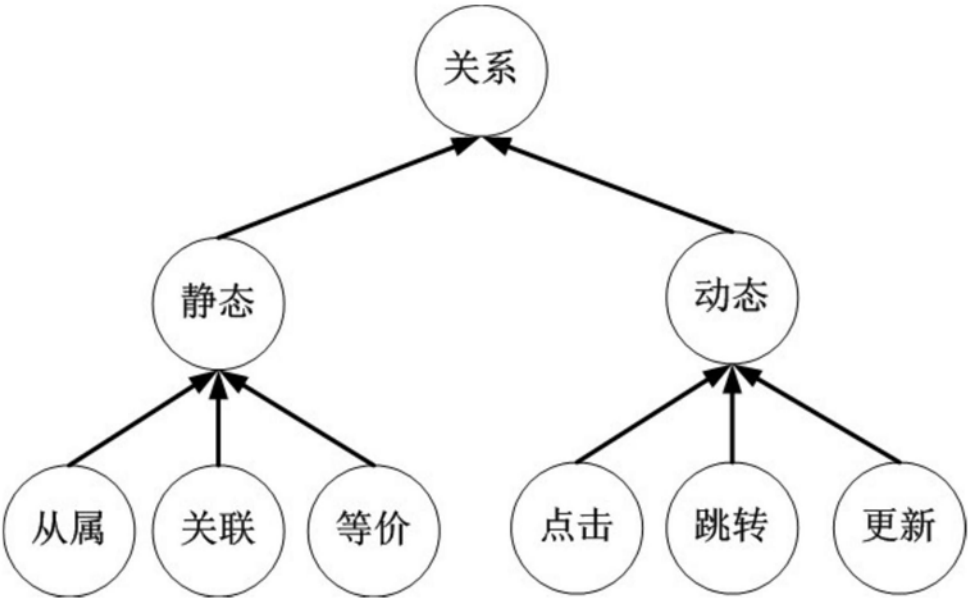


图3

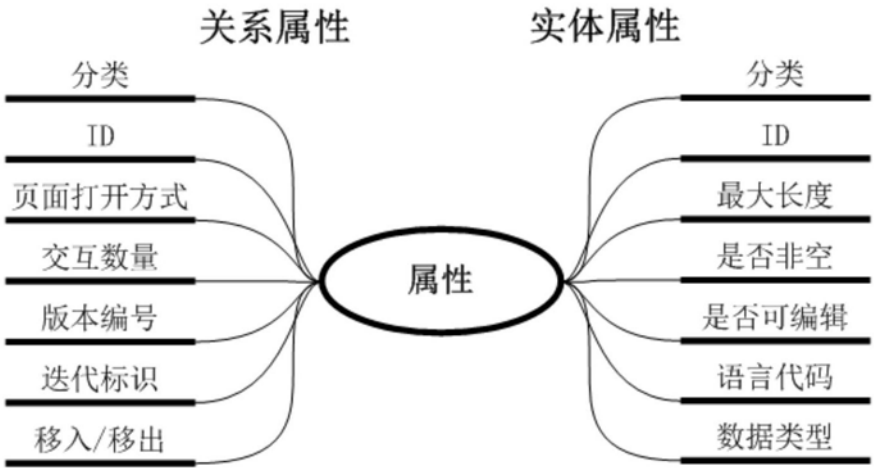


图4

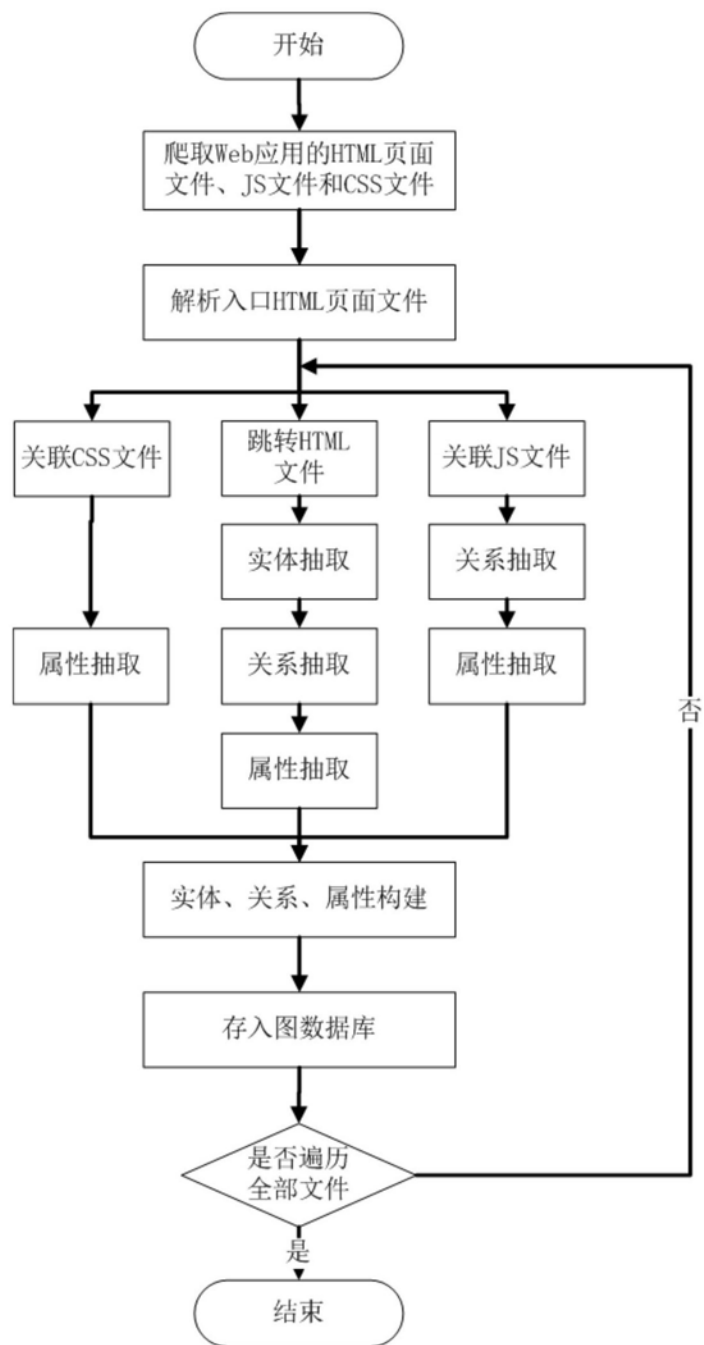


图5

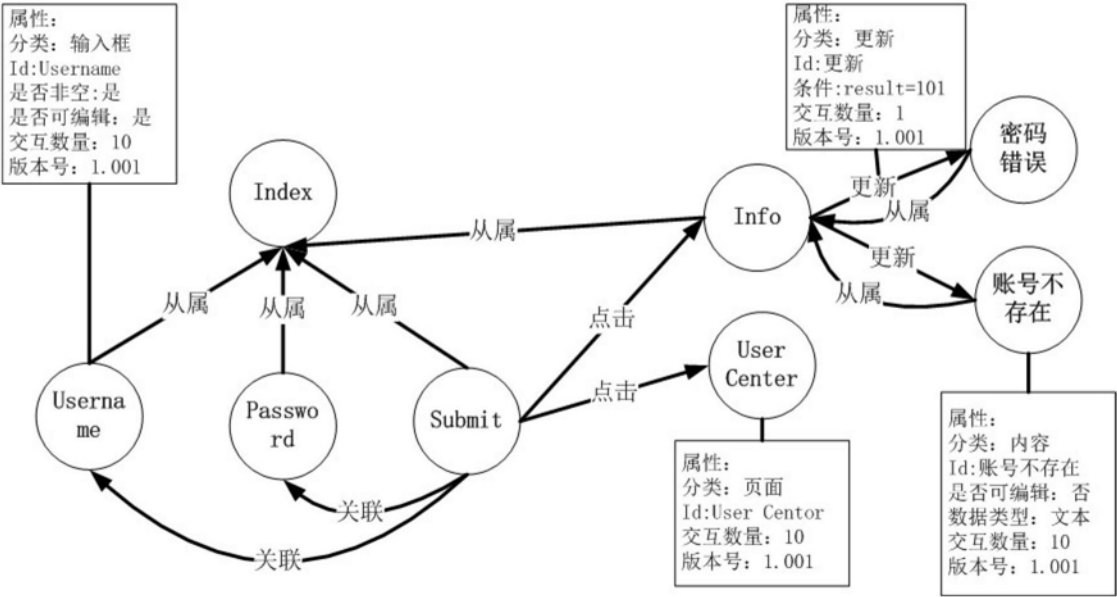


图6

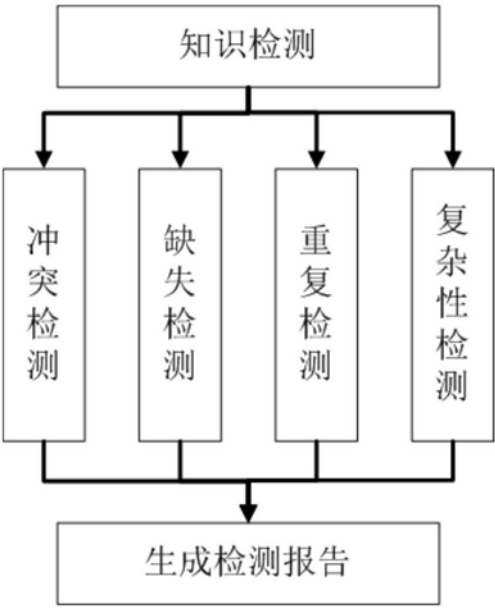


图7

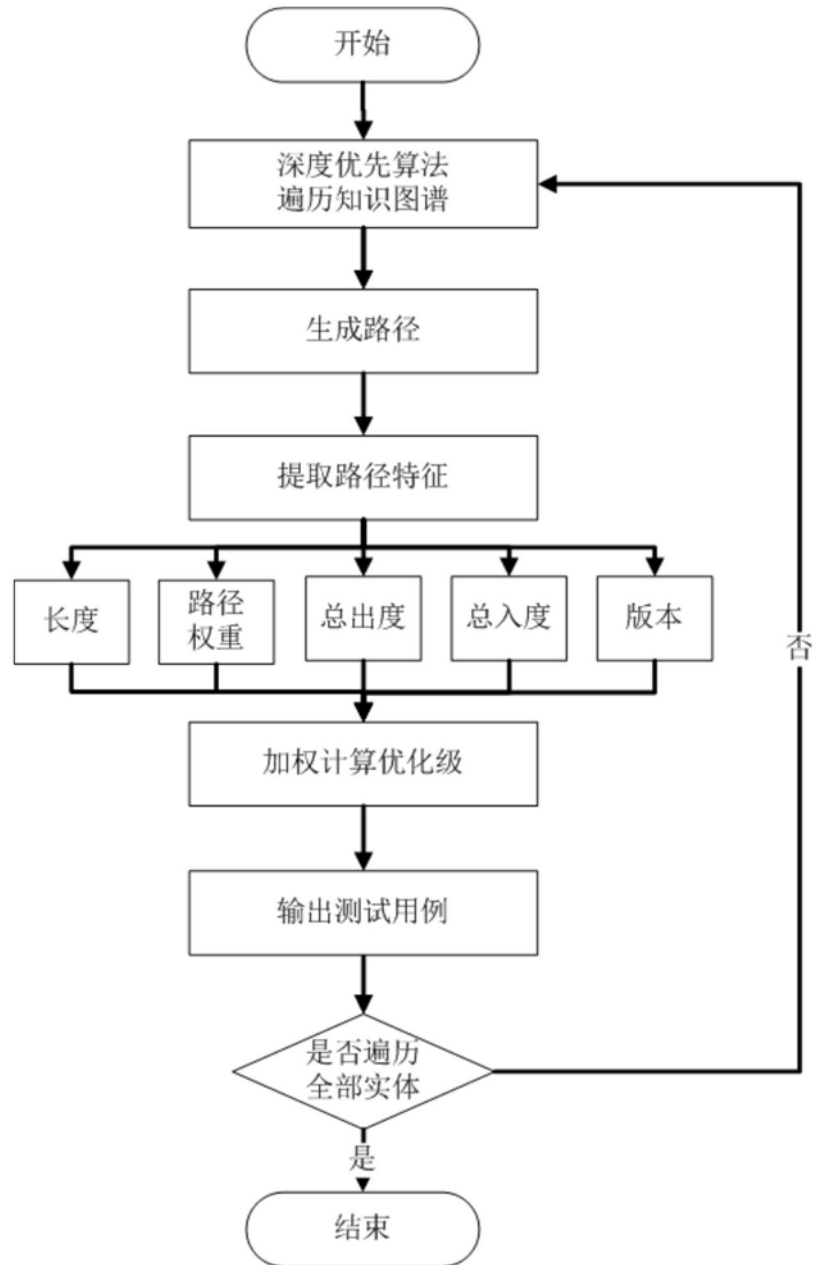


图8

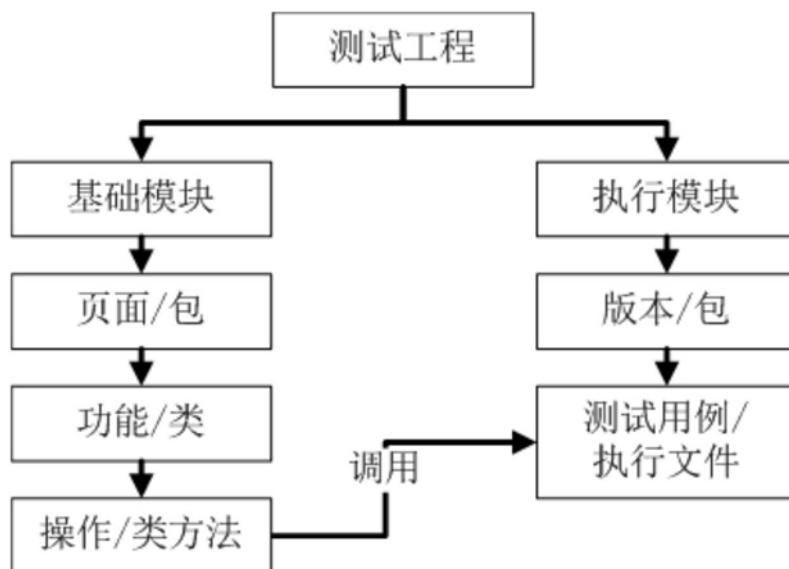


图9

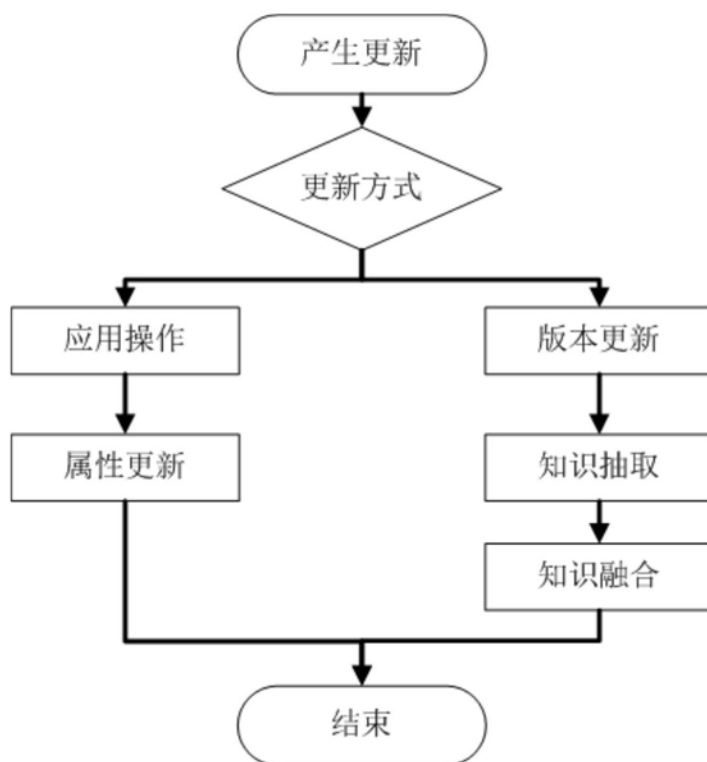


图10