



(12) 发明专利

(10) 授权公告号 CN 101984438 B

(45) 授权公告日 2012. 09. 05

(21) 申请号 201010561998. 4

(22) 申请日 2010. 11. 27

(73) 专利权人 杭州万格网络科技有限公司

地址 310051 浙江省杭州市滨江区滨盛路长河路口水印城 5 幢 2 单元 201 室

(72) 发明人 任浙东

(74) 专利代理机构 杭州中成专利事务所有限公司 33212

代理人 唐银益

(51) Int. Cl.

G06F 17/30 (2006. 01)

G06F 9/44 (2006. 01)

(56) 对比文件

CN 101751421 A, 2010. 06. 23, 说明书第 8, 12, 21, 22 段.

CN 101609399 A, 2009. 12. 23, 说明书第 2 页

第 4 段, 第 4 页第 1 段.

CN 101286120 A, 2008. 10. 15, 说明书第 3 页第 1-13 段, 第 6 页第 3 段 - 第 8 页第 6 段.

CN 101025738 A, 2007. 08. 29, 说明书第 1 页第 6 段 - 第 2 页第 5 段.

CN 101286120 A, 2008. 10. 15, 说明书第 3 页第 1-13 段, 第 6 页第 3 段 - 第 8 页第 6 段.

CN 101611399 A, 2009. 12. 23, 全文.

审查员 李燕东

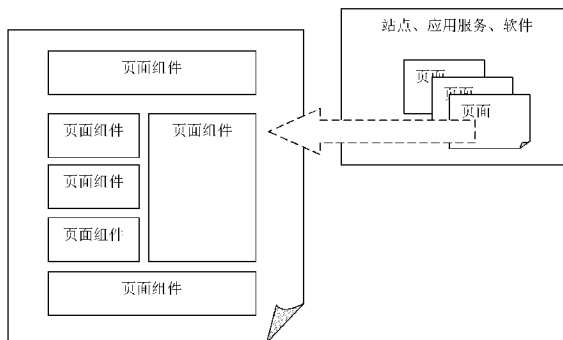
权利要求书 2 页 说明书 19 页 附图 10 页

(54) 发明名称

一种在线自助建站、创建应用服务与开发在线软件的方法

(57) 摘要

本发明涉及一种在线自助建站、创建应用服务与开发在线软件的方法。方法步骤一:在线开发页面、页面组件;步骤二:将页面创建为独立的站点、或应用服务、或在线软件。有益效果:为互联网在线自助建站、创建应用服务与开发在线软件提供了一套完整的方法,更是支持多用户,支持多语言,而且为开发 WEB 网站、WAP 网站、在线应用软件等等提供了一套统一的解决方案;独创了基于脚本语言及 XSL 语言的在线开发方法,为互联网在线应用软件开发模式带来一个全新的突破;提供了对象化管理数据的方法,规范了数据的定义,且引入了多语言的机制为数据标准化奠定基础;多用户的独立性,交互性,单点登录将互联网社区化走的更远。



1. 一种在线自助建站、创建应用服务与开发在线软件的方法,所述的站、应用服务、软件包括至少一个页面,所述的页面包括页面组件,其特征在于,方法如下:

步骤一:通过脚本语言将伺服端业务数据格式化输出成 XML 格式的数据,并将所述的数据通过 XSL 转换成页面语言,开发出页面、页面组件;

步骤二:将页面创建为独立的站点、或应用服务、或在线软件。

2. 根据权利要求 1 所述的方法,其特征在于,步骤一的实现方法是通过脚本语言及 XSL 开发页面及页面组件,方法如下:

1) 使用脚本语言及 XSL 可扩展样式表语言开发页面及页面组件的页面模板,页面模板以 XML 格式进行存储;

2) 使用页面模板引擎执行页面及页面组件的页面模板;

2.1) 使用脚本语言执行引擎执行页面模板上的脚本语言,将伺服端业务数据格式化输出成 XML 格式的数据;

2.2) 将 XML 数据通过页面模板上的 XSL 可扩展样式表语言转换成页面语言;

3) 页面与页面组件之间构成关联关系,在页面上操作页面组件,所述的一个页面包含至少一个以上页面组件;所述的一个页面组件可关联到至少一个以上页面;页面模板引擎执行页面的页面模板脚本代码时,使用页面模板引擎执行与页面关联的页面组件的脚本语言与 XSL 语言,并输出数据且作为整个页面 XML 数据源的一部分;

4) 通过页面组件直接对其关联的数据进行编辑,或者对其关联的数据模板自定义数据字段,页面组件获取数据的方法是:用脚本语言经过引擎格式化输出伺服端业务数据,步骤如下:

4.1) 对象数据管理(Object Data Manager)技术对业务数据存储系统的操作进行封装并向脚本语言提供可访问与操作业务数据的对象指针,其步骤如下:

4.1.1) 基于对象数据模板约束可操作的数据值类型、数据字段、枚举集合、多语言、对象数据之间的关联;

4.1.2) 对象化封装访问与操作业务数据;

4.1.3) 向脚本语言提供可访问与操作业务数据的对象指针;

4.1.4) 页面组件与对象数据模板之间构成关联关系。页面组件具有对象数据模板(pageportlet),被封装成对象;该对象数据模板(pageportlet)存有数据字段(entityTemplate),存储的值为关联的对象数据模板名称;

4.1.5) 页面与页面组件之间构成关联关系。页面也具有对象数据模板(page与pageportlet),被封装成对象;该对象数据模板(page)存有数据字段(portlets),存储的值为与之关联的页面组件编号及该组件在页面上的位置/层次信息;

4.2) 对象化封装访问与操作业务数据:把对象数据模板的访问与操作封装成对象(object),该对象拥有创建对象数据模板对应的存储表,对象数据也封装成对象;

4.3) 向脚本语言提供可访问与操作业务数据的对象指针,所述的指针包括封装对象数据模板操作的对象及其对象代理、封装对象数据操作的对象及其对象代理、查询器、过滤器、归组器,脚本代码通过所述指针访问与操作业务数据。

3. 根据权利要求 2 所述的方法,其特征在于,所述的页面模板开发可在客户端,开发好的页面模板可发送或保存至伺服端,所述的页面模板引擎执行是在伺服端。

4. 根据权利要求 2 所述的方法,其特征在于,所述的对象数据模板以 XML 格式进行存储。

5. 根据权利要求 1 所述的方法,其特征在于,步骤二的实现方法:通过跳转方式、链接路径的方式,或通过设置页面组件的关联配置,将所述的页面构成站。

6. 根据权利要求 5 所述的方法,其特征在于,通过访问页面以及页面之间的链接跳转完成对站的访问;通过通用逻辑事务访问页面。

7. 根据权利要求 6 所述的方法,其特征在于,所述的通用逻辑事务包括浏览页面、提交事务处理、载入文件、脚本文件执行、推送。

8. 根据权利要求 5 所述的方法,其特征在于,站及页面支持多语言,多语言可以下途径进行设置:

- 1) 页面的配置,强制指定当前页面使用的语言;
- 2) 针对访问用户的配置,强制指定访问者访问该站时使用的语言;
- 3) 页面请求的语言;
- 4) 客户端设置的语言;

所述的设置的优先级是逐级递减。

9. 根据权利要求 8 所述的方法,其特征在于,所述的语言信息记录在逻辑事务会话中。

10. 根据权利要求 1 所述的方法,其特征在于,所述的方法建立的多个站之间通过业务数据存储系统使用脚本语言进行数据共享。

11. 根据权利要求 10 所述的方法,其特征在于,所述的站之间的数据共享的操作权限采用组权限策略。

12. 根据权利要求 10 所述的方法,其特征在于,所述的一个站被设定为主站。

13. 根据权利要求 12 所述的方法,其特征在于,所述的站之间通过会话交换技术实现单点登录。

14. 根据权利要求 2 所述的方法,其特征在于,还可以使用页面语言转换程序可将页面语言转换成 PDF、SWF 格式。

一种在线自助建站、创建应用服务与开发在线软件的方法

技术领域

[0001] 本发明涉及自助建站、互联网开放平台、云计算、互联网在线应用服务及软件开发、软件及服务 SAAS,更具体地说,涉及一种在线自助建站、创建应用服务与开发在线软件的方法。

背景技术

[0002] 互联网创造了经济神话。企业网站、门户网站、电子商务网站、社区网站,手机 WAP 站点,移动互联网,客户服务管理系统 CRM、办公自动化系统 OA 等在线应用软件等等层出不穷,它们的增长速度都是任何预测家未曾预料到的。然而互联网在线开发的技术却没有得到快速的发展。现有市场上的自助建站平台也仅仅实现了简单的建站功能,无法满足用户需求多变万化的需求,无法实现多语言,而且也没有提出规范化的数据管理方法,仍然没有实质性的解决互联网数据孤岛的问题。

发明内容

[0003] 本发明的目的是解决以上提出的问题,提供一种多用户多语言在线自助建站、创建应用服务与开发在线软件的方法。

[0004] 本发明的技术方案是这样的:

[0005] 一种在线自助建站、创建应用服务与开发在线软件的方法,所述的站、应用服务、软件包括至少一个页面,所述的页面包括页面组件,方法如下:

[0006] 步骤一:通过脚本语言及 XSL 开发页面、页面组件;

[0007] 步骤二:将页面创建为独立的站点、或应用服务、或在线软件;

[0008] 步骤一的实现方法是通过脚本语言及 XSL 开发页面及页面组件,方法如下:

[0009] 1) 使用脚本语言及 XSL 可扩展样式表语言开发页面及页面组件的页面模板,页面模板以 XML 格式进行存储;

[0010] 2) 使用页面模板引擎执行页面及页面组件的页面模板;

[0011] 2.1) 使用脚本语言执行引擎执行页面模板上的脚本语言,将服务端业务数据格式化输出成 XML 格式的数据;

[0012] 2.2) 将 XML 数据通过页面模板上的 XSL 可扩展样式表语言转换成页面语言;

[0013] 3) 页面与页面组件之间构成关联关系,在页面上操作页面组件,所述的一个页面包含至少一个以上页面组件;所述的一个页面组件可关联到至少一个以上页面;页面模板引擎执行页面的页面模板脚本代码时,使用页面模板引擎执行与页面关联的页面组件的脚本语言与 XSL 语言,并输出数据且作为整个页面 XML 数据源的一部分;

[0014] 4) 通过页面组件直接对其关联的数据进行编辑,或者对其关联的数据模板自定义数据字段,页面组件获取数据的方法是:用脚本语言经过引擎格式化输出服务端业务数据,步骤如下:

[0015] 4.1) 对象数据管理 (Object Data Manager) 技术对业务数据存储系统的操作进行

封装并向脚本语言提供可访问与操作业务数据的对象指针,其步骤如下:

[0016] 4.1.1) 基于对象数据模板约束可操作的数据值类型、数据字段、枚举集合、多语言、对象数据之间的关联;

[0017] 4.1.2) 对象化封装访问与操作业务数据;

[0018] 4.1.3) 向脚本语言提供可访问与操作业务数据的对象指针;

[0019] 4.1.4) 页面组件与对象数据模板之间构成关联关系。页面组件具有对象数据模板 (pageportlet), 被封装成对象; 该对象数据模板 (pageportlet) 存有数据字段 (entityTemplate), 存储的值为关联的对象数据模板名称;

[0020] 4.1.5) 页面与页面组件之间构成关联关系。页面也具有对象数据模板 (page 与 pageportlet), 被封装成对象; 该对象数据模板 (page) 存有数据字段 (portlets), 存储的值为与之关联的页面组件编号及该组件在页面上的位置 / 层次信息;

[0021] 4.2) 对象化封装访问与操作业务数据: 把对象数据模板的访问与操作封装成对象 (object), 该对象拥有创建对象数据模板对应的存储表, 对象数据也封装成对象;

[0022] 4.3) 向脚本语言提供可访问与操作业务数据的对象指针, 所述的指针包括封装对象数据模板操作的对象以及其对象代理、封装对象数据操作的对象以及其对象代理、查询器、过滤器、归组器, 脚本代码通过所述指针访问与操作业务数据。

[0023] 作为优选, 所述的页面模板开发可在客户端, 开发好的页面模板可发送或保存至伺服器端, 所述的页面模板引擎执行是在伺服器端。

[0024] 作为优选, 所述的对象数据模板以 XML 格式进行存储。

[0025] 步骤二的实现方法: 通过跳转方式、链接路径的方式, 或通过设置页面组件的关联配置, 将所述的页面构成站。

[0026] 作为优选, 通过访问页面以及页面之间的链接跳转完成对站的访问; 通过通用逻辑事务访问页面。

[0027] 作为优选, 所述的通用逻辑事务包括浏览页面、提交事务处理、载入文件、脚本文件执行、推送。

[0028] 作为优选, 站及页面支持多语言, 多语言可以下途径进行设置:

[0029] 1) 页面的配置, 强制指定当前页面使用的语言;

[0030] 2) 针对访问用户的配置, 强制指定访问者访问该站时使用的语言;

[0031] 3) 页面请求的语言;

[0032] 4) 客户端设置的语言;

[0033] 所述的设置的优先级是逐级递减。

[0034] 作为优选, 所述的语言信息记录在逻辑事务会话中。

[0035] 作为优选, 所述的方法建立的多个站之间通过业务数据存储系统使用脚本语言进行数据共享。

[0036] 作为优选, 所述的站之间的数据共享的操作权限采用组权限策略。

[0037] 作为优选, 所述的一个站被设定为主站。

[0038] 作为优选, 所述的站之间通过会话交换技术实现单点登录。

[0039] 作为优选, 还可以使用页面语言转换程序可将页面语言转换成 PDF、SWF 格式。

[0040] 本发明的有益效果如下:

[0041] (1) 为互联网在线自助建站、创建应用服务与开发在线软件提供了一套完整的方法,更是支持多用户,支持多语言,而且为开发 WEB 网站、WAP 网站、在线应用软件等等提供了一套统一的解决方案。

[0042] (2) 独创了基于脚本语言及 XSL 语言的在线开发方法,为互联网在线应用软件开发模式带来一个全新的突破。

[0043] (3) 提供了对象化管理数据的方法,规范了数据的定义,且引入了多语言的机制为数据标准化奠定基础。

[0044] (4) 多用户的独立性,交互性,单点登录将互联网社区化走的更远。

[0045] (5) 开发者、用户、服务商,为互联网提供一种新的商务模式。

附图说明

[0046] 图 1 是站点、应用服务、软件的结构示意图;

[0047] 图 2 是伺服端业务数据格式化输出成 XML 格式的数据的示意图;

[0048] 图 3 是页面组件与页面之间的执行过程;

[0049] 图 4 是对象数据管理技术的实现方法的示意图;

[0050] 图 5 是 CSS 样式规则的示意图;

[0051] 图 6 是页面组件的部件的样式规范定义示意图;

[0052] 图 7 是通用逻辑事务的示意图;

[0053] 图 8 是站之间进行联系的示意图;

[0054] 图 9 是组权限策略的示意图;

[0055] 图 10 是各权限选项优先级以及依赖关系的示意图;

[0056] 图 11 是由页面及页面组件构成的网站的示意图;

[0057] 图 12 是组件管理菜单的示意图;

[0058] 图 13 是定制开发工具的界面的示意图;

[0059] 图 14 是对象数据模板的自定义的示意图;

[0060] 图 15 是对象数据的管理的示意图;

[0061] 图 16 是枚举集合的一种实施例——多选的示意图;

[0062] 图 17 是枚举集合的一种实施例——单选的示意图;

[0063] 图 18 是枚举集合的一种实施例——多级多选的示意图;

[0064] 图 19 是枚举集合的一种实施例——多级单选的示意图;

[0065] 图 20 是多语言站及页面的一个实施例——中文页面的示意图;

[0066] 图 21 是多语言站及页面的一个实施例——英文页面的示意图;

[0067] 图 22 是会话交换的一个实施例的示意图;

[0068] 图 23 是会话交换的另一个实施例的示意图。

具体实施方式

[0069] 下面结合附图对本发明的实施例进行进一步详细说明:

[0070] 一种在线自助建站、创建应用服务与开发在线软件的方法,所述的站、应用服务、软件包括至少一个页面,所述的页面包括页面组件,方法如下:

- [0071] 步骤一：在线开发页面、页面组件；
- [0072] 步骤二：将页面创建为独立的站点、或应用服务、或在线软件；
- [0073] 步骤一的实现方法是通过脚本语言及 XSL 开发页面及页面组件，方法如下：
- [0074] 1) 使用脚本语言及 XSL 可扩展样式表语言开发页面及页面组件的页面模板，页面模板以 XML 格式进行存储；
- [0075] 2) 使用页面模板引擎执行页面及页面组件的页面模板；
- [0076] 2.1) 使用脚本语言执行引擎执行页面模板上的脚本语言，将伺服端业务数据格式化输出成 XML 格式的数据；
- [0077] 2.2) 将 XML 数据通过页面模板上的 XSL 可扩展样式表语言转换成页面语言；
- [0078] 3) 页面与页面组件之间构成关联关系，在页面上操作页面组件，所述的一个页面包含至少一个以上页面组件；所述的一个页面组件可关联到至少一个以上页面；页面模板引擎执行页面的页面模板脚本代码时，使用页面模板引擎执行与页面关联的页面组件的脚本语言与 XSL 语言，并输出数据且作为整个页面 XML 数据源的一部分；
- [0079] 4) 通过页面组件直接对其关联的数据进行编辑，或者对其关联的数据模板自定义数据字段，页面组件获取数据的方法是：用脚本语言经过引擎格式化输出伺服端业务数据，步骤如下：
- [0080] 4.1) 对象数据管理 (Object Data Manager) 技术对业务数据存储系统的操作进行封装并向脚本语言提供可访问与操作业务数据的对象指针，其步骤如下：
- [0081] 4.1.1) 基于对象数据模板约束可操作的数据值类型、数据字段、枚举集合、多语言、对象数据之间的关联；
- [0082] 4.1.2) 对象化封装访问与操作业务数据；
- [0083] 4.1.3) 向脚本语言提供可访问与操作业务数据的对象指针；
- [0084] 4.1.4) 页面组件与对象数据模板之间构成关联关系。页面组件具有对象数据模板 (pageportlet)，被封装成对象。该对象数据模板 (pageportlet) 存有数据字段 (entityTemplate)，存储的值为关联的对象数据模板名称；
- [0085] 4.1.5) 页面与页面组件之间构成关联关系。页面也具有对象数据模板 (page 与 pageportlet)，被封装成对象。该对象数据模板 (page) 存有数据字段 (portlets)，存储的值为与之关联的页面组件编号及该组件在页面上的位置 / 层次信息；
- [0086] 4.2) 对象化封装访问与操作业务数据：把对象数据模板的访问与操作封装成对象 (object)，该对象拥有创建对象数据模板对应的存储表，对象数据也封装成对象；
- [0087] 4.3) 向脚本语言提供可访问与操作业务数据的对象指针，所述的指针包括封装对象数据模板操作的对象及其对象代理、封装对象数据操作的对象及其对象代理、查询器、过滤器、归组器，脚本代码通过所述指针访问与操作业务数据。
- [0088] 作为优选，所述的页面模板开发可在客户端，开发好的页面模板可发送或保存至伺服端。所述的页面模板引擎执行是在伺服端。
- [0089] 作为优选，所述的对象数据模板以 XML 格式进行存储。
- [0090] 步骤二的实现方法：通过跳转方式、链接路径的方式，或通过设置页面组件的关联配置，将所述的页面构成站。
- [0091] 作为优选，通过访问页面以及页面之间的链接跳转完成对站的访问；通过通用逻

辑事务访问页面。

[0092] 作为优选,所述的通用逻辑事务包括浏览页面、提交事务处理、载入文件、脚本文件执行、推送。

[0093] 作为优选,站及页面支持多语言,多语言可以下途径进行设置:

[0094] 1) 页面的配置,强制指定当前页面使用的语言;

[0095] 2) 针对访问用户的配置,强制指定访问者访问该站时使用的语言;

[0096] 3) 页面请求的语言;

[0097] 4) 客户端设置的语言;

[0098] 所述的设置的优先级是逐级递减。

[0099] 作为优选,所述的语言信息记录在逻辑事务会话中。

[0100] 作为优选,所述的方法建立的多个站之间通过业务数据存储系统使用脚本语言进行数据共享。

[0101] 作为优选,所述的站之间的数据共享的操作权限采用组权限策略。

[0102] 作为优选,所述的一个站被设定为主站。

[0103] 作为优选,所述的站之间通过会话交换技术实现单点登录。

[0104] 作为优选,还可以使用页面语言转换程序可将页面语言转换成 PDF、SWF 格式。

[0105] 如图 1 所示,一个 Web 网站、Flash 网站、WAP 站点或者在线应用软件等实际上都是由一系列页面构成的。同时,一个页面是由一系列页面组件组成的。页面组件即为实现一个特定功能需求的单元。在这里所指页面不局限于 WEB 网页,还包括 WAP 页面、应用服务界面、在线软件界面,以及各种终端设备在线访问浏览的客户端界面。页面是由语言来实现,在客户端被解析显示给用户界面。例如网页、Web 在线应用软件界面通常使用 HTML、XHTML 等语言。手机 WAP 站点通常使用 WML 语言。

[0106] 通过脚本语言及 XSL 开发页面及页面组件。页面组件虽然其特定的功能需求不一样,但其开发方式与页面的开发方式却都是一致的。

[0107] 如图 2 所示,一种被称之为页面模板的技术,其实现方法:

[0108] 1) 通过脚本语言将伺服器端业务数据格式化输出成 XML 格式的数据(称之为数据源)。这种脚本语言,用户是可以根据实际业务需要在线实时可编写的。为此我们提供了格式化脚本语言(GScript)以及脚本语言执行引擎(GScriptEngine)。

[0109] 2) 将 XML 数据源通过 XSL 可扩展样式表语言转换成页面语言。

[0110] 3) 页面语言也可经过转换程序(或称为引擎)转换成其他页面语言,本实施例中,可以转换成 PDF、SWF 格式。

[0111] 4) 页面模板(Page Template Engine)引擎调用脚本语言执行引擎(GScriptEngine)执行页面的脚本代码。

[0112] 5) 在执行页面的脚本代码时使用页面模板引擎执行每个页面组件的脚本语言与 XSL 语言,并输出数据且作为整个页面 XML 数据源的一部分。

[0113] 6) 页面模板引擎将页面的 XML 数据源经过 XSL 再转换成页面语言。

[0114] 步骤 4)、5)、6) 为页面组件与页面之间的执行过程,如图 3 所示。

[0115] 页面模板引擎包含三部分:

[0116] 1、脚本语言执行引擎;

- [0117] 2、XSL 转换 XML 数据源成页面语言；
- [0118] 3、页面语言转换成其他页面语言的程序或引擎。
- [0119] 页面模板引擎执行是在伺服器端。页面代码的开发可以在客户端，开发好的代码可发送或保存至伺服器端。页面代码包含以下两部分：
- [0120] 1、整个页面的脚本语言与 XSLT；
- [0121] 2、页面中每个页面组件的脚本语言与 XSLT。
- [0122] 我们已提供了浏览器在线开发工具。这工具在浏览器上就可以支持编写格式化脚本语言 (GScript)，页面模板 (Page Template)，JS，CSS，国际化语言文件 I18n(internationalization,本意是指软件的“国际化”)，XML 等。此极大的方便开发者随时随地开发页面及页面组件。
- [0123] 所述的在线开发工具，实现的方法如下：
- [0124] (1) 选定脚本语言 (GScript) 及脚本语言执行引擎 (GScriptEngine)；
- [0125] (2) 用户使用上述的语言根据业务需求编写脚本代码；
- [0126] (3) 客户端把脚本代码发送或保存到伺服器端；
- [0127] (4) 由伺服器端脚本语言执行引擎执行上述的脚本代码；
- [0128] (5) 执行的结果数据输出返回给客户端。
- [0129] 所述的执行引擎是格式化脚本语言执行引擎 (GScriptEngine)，用于接收传入的脚本代码，结合缓存会话 (context) 与请求 (request) 进行解析、执行代码，在解析执行过程中根据代码指令依次封装输出用户所需的数据。
- [0130] 所述的缓存会话 (context) 存储引用操作结构化 / 非结构化数据库的指针、引用操作文件 / 缓存数据 / 其它数据资源的指针、引用操作 CPU / 其它硬件设备的指针，以及存储在代码过程中根据用户指令临时产生的变量。
- [0131] 所述的请求 (request) 存储页面请求的参数名与参数值。
- [0132] 所述的格式化脚本语言执行引擎 (GScriptEngine) 入口参数主要包含脚本代码输入流 (inputstream)、缓存会话 (context)、请求 (request)、输出编码 (encoding)、结果数据输出流 (outputstream)、启用调试模式 (debug)。
- [0133] 所述的格式化脚本语言执行引擎解析代码的算法是变 - 分的方法，所述的变是指文本字符串不断的经过匹配替换而变成新的字符串，所述的分是指被替换的字符串进行分隔演变成多个新的字符串，算法按变 - 分 - 变 - 分 - 变循环处理。
- [0134] 其特征在于，所述的格式化脚本语言执行引擎处理代码的过程步骤如下：
- [0135] (1) 将脚本代码输入流转换成文本字符串；
- [0136] (2) 根据脚本指令函数语法变 - 分处理依次形成指令函数集合；
- [0137] (3) 按顺序依次执行指令函数；
- [0138] (4) 指令函数计算的结果根据指令函数的特性依次封装输出用户所需的数据。
- [0139] 所述的指令函数顺序为：先执行其函数参数集合，而后依次函数体内部的子指令函数。
- [0140] 指令函数参数集合执行时，对参数字符串进行分处理，依次分割成多个参数字符串，然后进行变处理。
- [0141] 所述的执行的结果数据的格式基于用户编写的脚本代码指令。

- [0142] 所述的业务数据的输出内容及格式可以是 XML、JSON、XML+XSL、XML+XSLT。
- [0143] 脚本代码发送或保存到伺服端的步骤是：由服务商提供入口，通过网络传送到伺服端，当客户端请求访问脚本代码时伺服端会自动调用脚本语言执行引擎对其解析、执行，并返回结果数据。
- [0144] 通过指令函数调用引用指针的属性或方法对业务数据进行操作，将执行的结果数据再通过指令函数格式化输出。
- [0145] 基于脚本语言扩展开放数据接口 (API)，可以通过扩展指令函数，或通过缓存会话 (context) 中引入对象指针的方式很容易扩展数据接口，或定义类名使用 assign 指令函数构造类对象。
- [0146] 脚本代码在伺服端的信任，采用现有公认的 PKI 安全体系及技术就可有效解决脚本代码在伺服端的信任问题。
- [0147] 伺服端解析执行页面模板引擎与脚本语言执行引擎的安全防护：
- [0148] (1) 引擎本身确保解析过程的安全，XPath 函数或指令函数执行的安全。
- [0149] (2) 脚本语言引用的对象指针可以使用安全策略文件限定可访问的属性与方法。
- [0150] (3) 页面模板引擎与脚本语言执行引擎。其所在的运行时环境例如 JAVA 语言其所需 JVM 虚拟机。可在运行时环境中加入安全策略。
- [0151] ■ 使用沙盒。让解析执行在受限的沙盒中运行。
- [0152] ■ 处理时长计时。控制解析执行的时间。达到限制时间后，处理将中断。
- [0153] ■ 限制脚本代码或页面模板的文件大小、语句数量。
- [0154] ■ 限制临时变量缓存的数量及大小，或者使用硬盘缓存策略。
- [0155] ■ 利用线程空闲及睡眠 sleep 控制 CPU 占用率。
- [0156] ■ 控制存在循环因素的 XSLT 对象，或指令函数被循环调用的次数。
- [0157] ■ 采用虚拟机。
- [0158] ■ 其它安全措施。
- [0159] (4) 监控解析执行情况，辅助安全防护。
- [0160] 用户开发好的或者配置好的页面与页面组件可保存，以便下次创建时引用或复制。同时也可以共享给其它用户，与其它用户分享页面或页面组件。
- [0161] 用户可快速引用或复制已开发好的页面或页面组件。这些可以是服务商提供的，也可以是自己开发的或配置好的，也可以使第三方开发者开发提供的。这些通常会被称为站点模板、页面模板、通用页面组件、定制页面组件、第三方页面组件等等。同时，(1) 用户可手工对这些组件进行配置。配置包括内容显示设置、组件特效例如漂浮、关联设置、样式定义等。当然也可以让系统自动调整设置。以方便用户快速设计一个漂亮美观实用的页面组件。(2) 用户可以二次开发这些组件。让个性化发挥到极致，适应满足各种需求。
- [0162] 用户可删除页面或者页面组件，也可从历史记录中恢复被删除掉的页面或页面组件。
- [0163] 用户在页面上就可自由通过鼠标拖拉或通过键盘调整各个组件各自所需位置 \ 大小 \ 上下显示层次。可见无需为页面创建布局，也无需编辑窗口或其它控制功能模块，就可以对页面进行排版。
- [0164] 所见即所得。页面上的组件添加、配置、拖拉调整、移除、二次开发等这些变化即与

在浏览时是实时保持一致的。

[0165] 如图 4 所示,通过页面组件可快速对其关联的数据进行编辑,或者对其关联的数据模板自定义数据字段。通常一个组件会与一种数据类型关联。例如用于显示最新新闻列表的与新闻 news 数据进行关联。页面组件获取数据的方法是用脚本语言经过引擎格式化输出服务端业务数据。使用脚本语言,用户就可以通过代码指令根据自己的意图格式化输出数据。同时也可操作业务数据。

[0166] 我们提供一种对象数据管理 (Object Data Manager) 技术对业务数据存储系统的操作进行封装并向脚本语言提供可访问与操作业务数据的对象指针,其步骤如下:

[0167] 1、基于对象数据模板约束可操作的数据值类型与数据字段

[0168] 2、对象化封装访问与操作业务数据

[0169] 3、向脚本语言提供可访问与操作业务数据的对象指针

[0170] 4、页面组件与对象数据模板之间构成关联关系。

[0171] 对象数据模板,以 XML 格式进行存储,用户可自定义数据模板。对象数据模板有时候我们简称为对象模板。其定义的数据字段我们称为对象属性。对象数据则被简称为对象。

[0172] 1、如何约束定义数据字段

[0173] (1)XML 定义规范该规范定义了:

[0174] ①版本号。

[0175] ②数据模板的别名,中英文。可扩展支持更多的语言。

[0176] ③定义数据的显示别名、访问与操作数据的权限约束、数据验证器、数据触发器、数据枚举。数据枚举用于指定该数据模板的某个数据字段的值是从枚举集合中选择的,单选或者多选。

[0177] ④定义该数据模板包含的数据字段,以及其名称、值类型、是否为关键字、中英文别名、默认值、描述(可扩展支持更多的语言)、最小值、最大值、最小长度、最大长度、值正则表达式、是否输入安全、是否独立语言。是否为关键字,一个数据模板只有一个为关键字。是否独立语言表示该数字字段的值是否随着不同的语言其值不同,还是无论何种语言其值都一直。是否输入安全表示该数字字段用户输入时是否隐藏输入的字符,例如密码输入。

[0178] (2) 自定义:

[0179] 用户可扩展自定义新的数据字段。以及覆盖定义数据的显示别名、访问与操作数据的权限约束、数据验证器、数据触发器、数据枚举等。

[0180] 2、数据值类型

[0181] 对象数据管理技术中支持的值类型有:

[0182] * 字符串型 string

[0183] a) 页面元素 Page Element 输入 INPUT

[0184] b) 最小长度 minLength 支持

[0185] c) 最大长度 maxLength 支持

[0186] d) 最小值 minValue 不支持

[0187] e) 最大值 maxValue 不支持

[0188] f) 值正则表达式 valueRegex 支持

[0189] g) 输入安全 inputSecurity 支持

[0190]	* 整型 integer		
[0191]	a) 页面元素	Page Element	输入 INPUT
[0192]	b) 最小长度	minLength	支持
[0193]	c) 最大长度	maxLength	支持
[0194]	d) 最小值	minValue	支持
[0195]	e) 最大值	maxValue	支持
[0196]	f) 值正则表达式	valueRegex	不支持
[0197]	g) 输入安全	inputSecurity	支持
[0198]	* 短整型 short		
[0199]	a) 页面元素	Page Element	输入 INPUT
[0200]	b) 最小长度	minLength	支持
[0201]	c) 最大长度	maxLength	支持
[0202]	d) 最小值	minValue	支持
[0203]	e) 最大值	maxValue	支持
[0204]	f) 值正则表达式	valueRegex	不支持
[0205]	g) 输入安全	inputSecurity	支持
[0206]	* 双浮点数 double		
[0207]	a) 页面元素	Page Element	输入 INPUT
[0208]	b) 最小长度	minLength	支持
[0209]	c) 最大长度	maxLength	支持
[0210]	d) 最小值	minValue	支持
[0211]	e) 最大值	maxValue	支持
[0212]	f) 值正则表达式	valueRegex	不支持
[0213]	g) 输入安全	inputSecurity	支持
[0214]	* 单浮点数 float		
[0215]	a) 页面元素	Page Element	输入 INPUT
[0216]	b) 最小长度	minLength	支持
[0217]	c) 最大长度	maxLength	支持
[0218]	d) 最小值	minValue	支持
[0219]	e) 最大值	maxValue	支持
[0220]	f) 值正则表达式	valueRegex	不支持
[0221]	g) 输入安全	inputSecurity	支持
[0222]	* 长整型 long		
[0223]	a) 页面元素	Page Element	输入 INPUT
[0224]	b) 最小长度	minLength	支持
[0225]	c) 最大长度	maxLength	支持
[0226]	d) 最小值	minValue	支持
[0227]	e) 最大值	maxValue	支持
[0228]	f) 值正则表达式	valueRegex	不支持

[0229]	g) 输入安全	inputSecurity	支持
[0230]	* 布尔型 boolean		
[0231]	a) 页面元素	Page Element	选择 SELECT
[0232]	b) 最小长度	minLength	
[0233]	0 或者 1, 小于 0 是为 0, 大于 0 时为 1		
[0234]	c) 最大长度	maxLength	不支持
[0235]	d) 最小值	minValue	不支持
[0236]	e) 最大值	maxValue	不支持
[0237]	f) 值正则表达式	valueRegex	不支持
[0238]	g) 输入安全	inputSecurity	不支持
[0239]	* 大文本 htext:string		
[0240]	a) 页面元素	Page Element	输入 TEXTAREA
[0241]	b) 最小长度	minLength	支持
[0242]	c) 最大长度	maxLength	支持
[0243]	d) 最小值	minValue	不支持
[0244]	e) 最大值	maxValue	不支持
[0245]	f) 值正则表达式	valueRegex	支持
[0246]	g) 输入安全	inputSecurity	不支持
[0247]	*HTML 大文本 htext:html		
[0248]	a) 页面元素	Page Element	输入 TEXTAREA
[0249]	b) 最小长度	minLength	支持
[0250]	c) 最大长度	maxLength	支持
[0251]	d) 最小值	minValue	不支持
[0252]	e) 最大值	maxValue	不支持
[0253]	f) 值正则表达式	valueRegex	支持
[0254]	g) 输入安全	inputSecurity	不支持
[0255]	* 文件夹 hfilefoder:hadoop		
[0256]	a) 页面元素	Page Element	输入 INPUT/ 选择 SELECT
[0257]	b) 最小长度	minLength	
[0258]	0 或者 1, 小于 0 是为 0, 大于 0 时为 1		
[0259]	c) 最大长度	maxLength	不支持
[0260]	d) 最小值	minValue	不支持
[0261]	e) 最大值	maxValue	不支持
[0262]	f) 值正则表达式	valueRegex	不支持
[0263]	g) 输入安全	inputSecurity	不支持
[0264]	* 文件 hfile:hadoop		
[0265]	a) 页面元素	Page Element	输入 INPUT/ 选择 SELECT
[0266]	b) 最小长度	minLength	
[0267]	0 或者 1, 小于 0 是为 0, 大于 0 时为 1		

- | | | | |
|--------|--|---------------|---------------------|
| [0268] | c) 最大长度 | maxLength | 不支持 |
| [0269] | d) 最小值 | minValue | 不支持 |
| [0270] | e) 最大值 | maxValue | 不支持 |
| [0271] | f) 值正则表达式 | valueRegex | 不支持 |
| [0272] | g) 输入安全 | inputSecurity | 不支持 |
| [0273] | * 编号值类型为长整型 long 的对象类型 hthing:long | | |
| [0274] | a) 页面元素 | Page Element | 输入 INPUT/ 选择 SELECT |
| [0275] | b) 最小长度 | minLength | |
| [0276] | 0 或者 1, 小于 0 是为 0, 大于 0 时为 1 | | |
| [0277] | c) 最大长度 | maxLength | 不支持 |
| [0278] | d) 最小值 | minValue | 不支持 |
| [0279] | e) 最大值 | maxValue | 不支持 |
| [0280] | f) 值正则表达式 | valueRegex | 不支持 |
| [0281] | g) 输入安全 | inputSecurity | 不支持 |
| [0282] | * 编号值类型为字符串 string 的对象类型 hthing:string | | |
| [0283] | a) 页面元素 | Page Element | 输入 INPUT/ 选择 SELECT |
| [0284] | b) 最小长度 | minLength | |
| [0285] | 0 或者 1, 小于 0 是为 0, 大于 0 时为 1 | | |
| [0286] | c) 最大长度 | maxLength | 不支持 |
| [0287] | d) 最小值 | minValue | 不支持 |
| [0288] | e) 最大值 | maxValue | 不支持 |
| [0289] | f) 值正则表达式 | valueRegex | 不支持 |
| [0290] | g) 输入安全 | inputSecurity | 不支持 |
| [0291] | * 指定对象模板的对象类型 hthing:template#< 对象数据模板名称 > | | |
| [0292] | a) 页面元素 | Page Element | 输入 INPUT/ 选择 SELECT |
| [0293] | b) 最小长度 | minLength | |
| [0294] | 0 或者 1, 小于 0 是为 0, 大于 0 时为 1 | | |
| [0295] | c) 最大长度 | maxLength | 不支持 |
| [0296] | d) 最小值 | minValue | 不支持 |
| [0297] | e) 最大值 | maxValue | 不支持 |
| [0298] | f) 值正则表达式 | valueRegex | 不支持 |
| [0299] | g) 输入安全 | inputSecurity | 不支持 |
| [0300] | * 日期型 htime:date | | |
| [0301] | a) 页面元素 | Page Element | 输入 INPUT/ 选择 SELECT |
| [0302] | b) 最小长度 | minLength | |
| [0303] | 0 或者 1, 小于 0 是为 0, 大于 0 时为 1 | | |
| [0304] | c) 最大长度 | maxLength | 不支持 |
| [0305] | d) 最小值 | minValue | 特殊定义, 参见后面 |
| [0306] | e) 最大值 | maxValue | 特殊定义, 参见后面 |

- [0307] f) 值正则表达式 valueRegex 不支持
- [0308] g) 输入安全 inputSecurity 不支持
- [0309] * 时间型 htime:timestamp
- [0310] a) 页面元素 Page Element 输入 INPUT/ 选择 SELECT
- [0311] b) 最小长度 minLength
- [0312] 0 或者 1, 小于 0 是为 0, 大于 0 时为 1
- [0313] c) 最大长度 maxLength 不支持
- [0314] d) 最小值 minValue 特殊定义, 参见后面
- [0315] e) 最大值 maxValue 特殊定义, 参见后面
- [0316] f) 值正则表达式 valueRegex 不支持
- [0317] g) 输入安全 inputSecurity 不支持
- [0318] * 对象模板类型 hthingbase:template
- [0319] a) 页面元素 Page Element 选择 SELECT
- [0320] b) 最小长度 minLength
- [0321] 0 或者 1, 小于 0 是为 0, 大于 0 时为 1
- [0322] c) 最大长度 maxLength 不支持
- [0323] d) 最小值 minValue 不支持
- [0324] e) 最大值 maxValue 不支持
- [0325] f) 值正则表达式 valueRegex 不支持
- [0326] g) 输入安全 inputSecurity 不支持
- [0327] * 整型的 XYZWH 形式的空间位置类型 hspace:xyzwh_int
- [0328] 该类型标定了 X 轴, Y 轴, Z 轴的位置。考虑到 Web 页面的二维的特性, Z 轴数值是对应到 CSS 样式中的 z-index 值。在 Web 页面中 X 轴的值以 1024 为基准, Y 轴的值是 768 为基准。另外 X 轴与 Y 轴还有两个对应的数值 xPercent 与 yPercent, 用于描述当前 X 轴的值与 Y 轴的值是否是百分比形式的。如果是百分比形式则在 Web 页面展示时须要乘以基准才能得到实际值。
- [0329] 该类型还标定了宽度 W 与高度 H。在 Web 页面中宽度 W 以 X 轴的基准为基准, Y 轴的值是以 Y 轴的基准为基准。同样还有两个对应的数值 wPercent 与 hPercent, 用于描述宽度 W 与高度 H 的值是否是百分比形式的。如果是百分比形式则在 Web 页面展示时须要乘以基准才能得到实际值。
- [0330] * 整型的 RCWH 形式的空间位置类型 hspace:rcwh_int
- [0331] 该类型标定了行 R (Row) 与列 C (Column)。在 Portal 布局中使用行列标定位置的。
- [0332] 该类型还标定了宽度 W 与高度 H。在 Web 页面中宽度 W 以 1024 为基准, Y 轴的值是以 768 为基准。同样还有两个对应的数值 wPercent 与 hPercent, 用于描述宽度 W 与高度 H 的值是否是百分比形式的。如果是百分比形式则在 Web 页面展示时须要乘以基准才能得到实际值。
- [0333] * 实体状态型 htime:entitystate
- [0334] 用于描述实体所处的状态。运用实体所处的状态, 将其作为该关系存在时间类型。
- [0335] a) 容器状态显示时' container.state = VIEW'

[0336] 用于对象 page 与对象 pageportlet 之间。

[0337] b) 任意时 '*'

[0338] 用于被引用与引用类关系。

[0339] 对于日期型 htime:date、时间型 htime:timestamp 的最小值 minValue 与最大值 maxValue 如下特殊定义：

[0340] (1) 如果在大于 -1 小于 1 且不为 0 情况下 0.<增量><最后一位> 如下 2-9 项进行处理。否则为当前日期或创建日期 (htime:date) 或当前时间或创建时间 (htime:timestamp)。

[0341] (2) 如果<最后一位>为 1, 则<增量>计算为日, 以当前日期或创建日期 (htime:date) 或当前时间或创建时间 (htime:timestamp) 为基准类推增量计算所得, 例如 0.11 表示累加 1 日。

[0342] (3) 如果<最后一位>为 2, 则<增量>计算为月, 同上类推增量计算所得。

[0343] (4) 如果<最后一位>为 3, 则<增量>计算为年, 同上类推增量计算所得。

[0344] (5) 如果<最后一位>为 4, 则<增量>计算为时, 同上类推增量计算所得。

[0345] (6) 如果<最后一位>为 5, 则<增量>计算为分, 同上类推增量计算所得。

[0346] (7) 如果<最后一位>为 6, 则<增量>计算为秒, 同上类推增量计算所得。

[0347] (8) 如果<最后一位>为 7, 则<增量>计算为毫妙, 同上类推增量计算所得。

[0348] (9) 如果<最后一位>为 8 或者 9, 无效。

[0349] 另外值类型还有两种复杂的类型, 复合类型与结构类型。

[0350] 复合类型 (complex) 必须由简单类型 (以上除 hspace:xyzwh_int, hspace:rcwh_int 外的所有类型) 构成。复合类型定义方法为 complex(<名称 1><类型 1><是否与语言无关 false|true><是否安全输入 false|true><最小长度 NULL|数字><最大长度 NULL|数字><语言 1 默认值|语言 2 默认值|...>, <名称 2><类型 2><是否与语言无关 false|true><是否安全输入 false|true><最小长度 NULL|数字><最大长度 NULL|数字><语言 1 默认值|语言 2 默认值|...>, ...)

[0351] 结构类型 (structure) 由简单类型或复合类型构成。其定义方法为 :structure {<名称 1><类型 1><是否与语言无关 false|true><是否安全输入 false|true><最小长度 NULL|数字><最大长度 NULL|数字><语言 1 默认值|语言 2 默认值|...>; <名称 1><类型 1><是否与语言无关 false|true><是否安全输入 false|true><最小长度 NULL|数字><最大长度 NULL|数字><语言 1 默认值|语言 2 默认值|...>; complex(<名称 1><类型 1><是否与语言无关 false|true><是否安全输入 false|true><最小长度 NULL|数字><最大长度 NULL|数字><语言 1 默认值|语言 2 默认值|...>, <名称 2><类型 2><是否与语言无关 false|true><是否安全输入 false|true><最小长度 NULL|数字><最大长度 NULL|数字><语言 1 默认值|语言 2 默认值|...>, ...); ...}。

[0352] 3、枚举集合

[0353] 以 XML 格式进行存储, 用户可自定义或重载枚举集合。用户也可对其中一枚举集合自定义增加枚举项。其增加的方式就是重载枚举集合的方式。

[0354] (1) XML 定义规范, 该规范定义了：

[0355] ①该枚举集合的名称；

- [0356] ②该枚举集合是否允许用户重载定义；
- [0357] ③该枚举集合的应用范围 scope；
- [0358] ④中英文枚举集合别名,以及枚举项;可扩展支持更多的语言；
- [0359] a) 枚举项的值 (key)；
- [0360] b) 枚举项的文本 (text)；
- [0361] c) 枚举项的反向文本 (reverse-text)。反向文本是从不同角度去看。例如个人输入性别时会选择保密,而在服务商看来该性别文本不是保密而是未知。
- [0362] (2) 数据模板中如何引用枚举集合
- [0363] 在数据模板中通过关联数据字段方式与枚举集合关联。关联时可支持单选与多选。
- [0364] ①一对一关联,例如性别。
- [0365] ②一对多关联,例如区域包含省份、城市。
- [0366] 4、多语言
- [0367] 可扩展支持更多的语言。当数据字段标定为多语言时则该数据字段的值可根据不同的语言进行存储。
- [0368] 5、对象数据之间的关联
- [0369] 对象数据的关联,类似于 SQL 关系数据库中的主键与外键,系统使用了特殊的值类型方式进行关联。这些可关联的值类型有：
- [0370] (1) 编号值类型为长整型 long 的对象类型 hthing:long。
- [0371] (2) 编号值类型为字符串 string 的对象类型 hthing:string。
- [0372] (3) 指定对象模板的对象类型 hthing:template#< 对象数据模板名称 >。
- [0373] (4) 文件夹 hfilefoder:hadoop。其关联到文件夹 filefolder 对象数据模板。
- [0374] (5) 文件 hfile:hadoop。其关联到文件 file 对象数据模板。
- [0375] 对象化封装访问与操作业务数据。我们把对象数据模板的访问与操作封装成对象 (object)。该对象拥有创建对象数据模板对应的存储表,销毁存储表,添加与删除对象数据,添加、更新、删除数据字段,使用查询器 query、过滤器 filter、归组器 group 查询数据等的功能。我们把每一条对象数据也封装成对象。该对象则拥有访问与更新数据的功能。通过对象化就很容易将操作对象数据模板与对象数据的日志记录保留下来,为用户提供历史追溯,恢复与还原数据。
- [0376] 如是页面与页面组件各自都具有对象数据模板 (page 与 pageportlet),也都被封装成对象。其中页面组件的对象数据模板存在一个数据字段 (entityTemplate),其值类型为 hthingbase:template,存储的值就是对象数据模板名称。从而使得页面组件与对象数据模板之间构成关联关系。另外还有一个数据字段 (entityIdentity),其值类型为 hthing:string,存储的值为对象数据的编号 (id)。当该值存在时表示该页面组件与该对象数据之间构成关联关系。
- [0377] 向脚本语言提供可访问与操作业务数据的对象指针。这些指针包括封装对象数据模板操作的对象以及其对象代理、封装对象数据操作的对象以及其对象代理、查询器、过滤器、归组器等等。通过这些指针,脚本代码就非常容易访问与操作业务数据。
- [0378] 页面的语言代码规范,防止跨站脚本攻击 (XSS,又叫 CSS(Cross Site Script))。

它指的是恶意攻击者往 Web 页面里插入恶意 html 代码,当用户浏览该页之时,嵌入其中 Web 里面的 html 代码会被执行,从而达到恶意攻击用户的特殊目的。)注入漏洞,对客户端加载的性能优化,以及针对第三方商业服务例如搜索引擎优化 (SEO, search engine optimization,) 的优化等等。例如页面上 js 脚本文件、css 样式文件进行了压缩 (YUI compress),以提高浏览器加载页面的速度。

[0379] CSS 样式定义的规范方法,该规范的方法在于:

[0380] (1) 归类构成页面组件的部件,例如头部 header、列表 list、表格 table、分页 pagination。

[0381] (2) 对部件逐个做样式规范定义。

[0382] (3) 允许一个部件存在着多种形式,但这些形式都遵守其样式规范定义。

[0383] (4) 开发页面组件时就组合各种部件,并引用这些部件的样式。也就是说组件的样式是有这些部件的样式构成的。例如新闻列表的样式其由箱体 box、头部 header、列表 list、表格 table、分页 pagination 等的样式构成。

[0384] 我们把组件主体划分为 box 箱体、list 列表、pagination 分页、block 内容块等多个部件。其中 box 箱体必须存在,其分为三部分 :header、body、footer。如下所述:

[0385]

```
.box {}
.box .header {}
.box .body {}
.box .footer {}

<div class="box">

<div class="header">Company Introduce</div>
<div class="body">
    Company Introduce.....Company Introduce.....
</div>
<div class="footer"></div>
</div>
```

[0386] box 又根据组件不同分别划分为不同的组件 box 样式。例如 company 公司介绍,可以存在多种 company-box, company-box-1, company-box2。

[0387] list 等其它形式的则不具备根据组件不同进行划分,但可以存在多种例如 list, list-1, list-2 如附带的 list.html 页面。

[0388]

```

<!--base -->
.box {}
.box .header {}
.box .body {}
.box .footer {}

<!--company -->
.company-box {
height:100%;
width:100%;
border: 1px solid red;
font-size:16px;
}
.company-box .header {
font-weight:bold;
}
.company-box .body {}
.company-box .footer {}

<!--block -->
.block-1 {color:gray;}

<div class="box company-box">
<div class="header">Company Introduce</div>
<div class="body block-1">

```

[0389]

```

    Company Introduce.....Company Introduce.....
</div>
<div class="footer"></div>
</div>

```

[0390] 如上样式规则,如图5所示,原理是 company-box 系列定义的 header, body, footer 根据顺序会覆盖 box 定义的,而 block-1 系列样式定义会与 company-box body, boxbody 叠加样式。根据这个机制,我们就可以如是处理 list 了,如图6所示,我们已经对箱体 box、头部 header、文本 text、表格 table、幻灯片 slide、滚动条 scroll、分页 pagination、浮动 overlay、导航栏 navigator、列表 list、窗体 form、尾部 footer、元素 component、logo、homepage、favorite、TAB 标签 tabs 等部件做出了规范定义。

[0391] 页面的语言代码规范,防止 XSS 入注漏洞,对客户端加载的性能优化,以及针对第三方商业服务例如 SEO 的优化等等。例如页面上 js 脚本文件、css 样式文件进行了压缩 (YUI compress),以提高浏览器加载页面的速度。

[0392] 每个用户通过一系列页面创建独立的站点,或应用服务,或在线软件。这里我们将站点,或应用服务,或在线软件都统一简称为站。

[0393] 页面之间的跳转方式、链接路径可在开发中显示指定,也可通过页面组件的关联配置来进行设置。这样,这些页面就构成了一个完整的站。站的访问是通过访问页面以及

页面之间的链接跳转完成的。

[0394] 页面的访问方法。页面包括脚本文件 (GScript) 等都是通过通用逻辑事务 (Action) 进行访问的。通用逻辑事务通常根据需要进行划分, 浏览页面 (VIEW), 提交事务处理 (POST), 载入文件 (LOAD), 脚本文件执行 (GSCRIPT), 推送 (PUSHLET) 等, 如图 7 所示。这些逻辑事务类似于传统的通用网关接口 (CGI, Common Gateway Interface)。

[0395] 逻辑事务在执行时会创建事务会话 (ActionContext)。该事务会话仅在当前请求的过程中有效, 当请求完毕后即刻销毁。在脚本代码 (GScript) 中引用该事务会话, 即可间接性的引用到用户登录会话 (Session Context), 请求会话 (Request Context), 响应会话 (Response Context), 脚本会话 (JS Context), 样式会话 (CSS Context), 安全会话 (Policy Context) 等。用户的登录信息是在用户登录会话 (Session Context) 中记录; 用户访问的页面信息及请求参数是在请求会话 (Request Context) 记录; 脚本会话 (JS Context) 记录了被加载的 JS 脚本文件; 样式会话 (CSS Context) 记录了被加载的 CSS 样式文件; 等等。

[0396] 页面的参数传递。页面的参数传递遵从 URL 或窗体 FORM 参数的定义规则。根据参数的有效范围被划分成两种, 一是页面组件级的参数, 其参数名包含了标定页面组件的特征信息, 例如通常为页面组件编号加上 - 而后跟上参数名, 类似于: < 页面组件编号 >-< 参数名 >。这样在页面组件脚本开发时就能有效获取与当前组件相关的参数。二是页面级的参数, 其参数名不包含标定页面组件的特征信息。这些参数被请求后经过逻辑事务自动记录到请求会话 (Request Context)。在脚本开发时引用该对象指针即可获得参数信息。另外, 为防止 XSS 注入漏洞, 通常参数值会经过 XSS 的过滤器, 以避免恶意请求带来的安全隐患。

[0397] 站及页面的多语言。多语言可通过多种途径进行设置:

[0398] 1、页面的配置。强制指定当前页面使用的语言。

[0399] 2、针对访问用户的配置。强制指定访问者访问该站时使用的语言。

[0400] 3、页面请求的语言。

[0401] 4、客户端设置的语言。

[0402] 这些设置的优先级是从上往下降低。语言信息记录在逻辑事务会话 (ActionCojntext) 中。在脚本开发时很容易获得当前事务使用的语言。通过获取语言, 再加上数据对多语言的支持, 让开发者非常容易开发多语言页面或站。

[0403] 站之间的联系。站之间相互独立, 但是站之间又能互相共享。如图 8 所示, 一个用户站试图去访问或操作另外一个用户站的数据时候需要受到另外一个用户站对其设置的权限所约束。用户站之间的数据共享仍然是通过脚本语言进行的。脚本语言所引用的对象数据模板操作指针具备访问其它用户站的对象数据的功能, 但访问时受到其它用户站对当前用户站的权限约束。该约束是由系统来完成的。

[0404] 我们提供了一种权限约束规则:

[0405] 操作权限设计采用组权限策略。用户必须属于站的某个权限组, 否则无论是否登录都被认为是匿名用户。我们认为拥有站 A 的用户是站 A 的拥有者。也就是站 A 是该拥有者的自己站。而访问用户是指访问该站的用户, 可能是该站的注册用户, 也有可能是一个游客。

[0406] 如图 9 所示, 访问用户访问站 A, 权限控制依赖于用户在站 A 所在的组权限。当站

A 访问站 B 数据时, 权限控制依赖于站 A 的拥有者在站 B 所在的组权限, 并不会因为访问用户变化而发生变化。站 B 向站 A 开放数据操作权限时, 将无法约束访问用户通过访问站 A 间接的操作这些数据。

[0407] 安全权限级别划分为两级: (1) 系统级 (PLATFORM), 由系统代理授权处理。(2) 用户级, 根据用户组权限进行授权处理。其中用户级又划分为两级, 一为普通级 GENERAL, 该级别主要由权限对象控制。在事务处理过程中可授权忽略这些权限控制, 从而能够更灵活的满足应用需求。二是私密级 (PRIVATE), 该级别主要由权限配置文件控制。除用户自身可突破该配置文件配置的权限控制外, 其他人无论如何被授权均必须受这些配置所约束。通常用于用户保护私有数据, 以阻止授权开发的人员试图通过代码访问到这些数据。

[0408] 当用户访问自己站时则会自动忽略私密级 (PRIVATE) 与普通级 (GENERAL) 的安全限制。所以拥有者自身无论所处哪个群组都不受权限对象控制。

[0409] 所述的系统级 (PLATFORM)、私密级 (PRIVATE)、普通级 (GENERAL) 都是我们在权限控制上定义的特定的标示符。

[0410] 安全权限控制的优先策略如下:

[0411] i. “访问用户——站 A——站 B”

[0412] 用户访问站 A 是以用户在站 A 的身份进行判断权限, 也就是由站 A 控制。站 A 访问站 B 的数据则以站 A 在站 B 的身份进行判断权限, 也就是说站 B 控制。访问用户到站 A, 站 A 可忽略当前站用户普通级 GENERAL 的权限控制。但是站 A 到站 B, 站 A 是无法忽略站 B 的用户普通级 GENERAL 的权限控制。

[0413] ii 用户的权限控制优先于组的权限控制

[0414] iii. 对象数据 (对象) 的权限控制优先于对象数据模板 (对象模板) 的权限控制

[0415] iv. 拒绝优先于允许

[0416] 只要被拒绝则就认为被拒绝。若被允许, 则仍然需要继续判断下一个是否被允许或拒绝。

[0417] v. 多个组合的检查项, 只要其中一个检查项被拒绝则就拒绝。例如访问对象数据模板 (accessEntities), 访问对象数据 (objectAccess)。访问对象数据 (objectAccess) 依存于访问对象数据模板 (accessEntities), 所以只要其中一项被拒绝, 则就拒绝, 其检查项如图 10 所示。

[0418] 所述的对象数据模板 (accessEntities), 访问对象数据 (objectAccess) 是在权限控制上定义的特定的标示符

[0419] 安全权限与访问来源 (referer, 标准英文单词为 referrer)。通用逻辑事务中有些事务例如提交事务处理 (POST) 与访问来源有关, 如果访问来源与当前站不在同一个域时是被拒绝执行的。

[0420] 单点登录 (SSO, Single Sign On。SSO 是在多个应用系统中, 用户只需要登录一次就可以访问所有相互信任的应用系统)。设定其中一个站为主站, 通常将服务商的站设为主站。用户在这些站之间通过会话交换技术 (Session Switch) 即可实现单点登录 SSO。服务商是指为用户提供在线自助建站、创建应用服务与开发在线软件的提供商。其方法是利用浏览器的跨域安全限制, 实现用户从一个站 (称为来源) 登录到另一个站 (称为当前站) 时的自动切换登录。

[0421] 自动切换登录的基准如下：

[0422] 1、当前站是主站，则以来源为基准进行切换登录。

[0423] 2、果当前站不是主站，则以当前站为基准进行切换登录。

[0424] 切换原则是始终不能以主站为基准。

[0425] 在来源为基准下，(1) 若来源已经登录则判断来源是否为访问者自己的站，(a) 若是，访问者则可选择要么输入验证码切换登录到主站要么继续访问此页面。(b) 若不是，则显示是否登录主站。(2) 若来源未登录，则显示是否登录主站，与登录此站。

[0426] 在当前站为基准下，(1) 若已经登录主站，则访问者可选择要么输入验证码切换登录到当前站要么继续访问此页面。仅在登录主站的用户与当前站的拥有者（表示访问者正在登录自己的站）一致时才会自动登录。(2) 若主站没有登录则显示是否登录到主站，与登录此站。

[0427] 实施例

[0428] 图 11 所示，图中，1 是 HTML 框，2 是导航栏，3 是多语言，4 是 FLASH 框，5 是公司介绍，6 是图片框，7 是图片框。图 1 至图 7 都是页面组件，共同构成页面。

[0429] 图 12 所示的是组件管理菜单。

[0430] 图 13 所示的是定制开发工具的界面，图中，9 是支持的语言，10 是 XML 数据源，11 是输出区，12 是调试区，13 是代码区。

[0431] 图 14 所示的是对象数据模板的自定义操作界面；

[0432] 图 15 所示的是对象数据的管理界面；

[0433] 图 16、图 17、图 18、图 19 所示的是枚举集合几种类型，分别是多选、单选、多级多选、多级单选。

[0434] 图 20、图 21 所示的是多语言站及页面，页面请求语言分别是中文、英文。

[0435] 图 22、图 23 所示的是会话交换的几种类型，分别包括登录主站后访问者自己的站、未登录主站时访问平台或站、登录主站后访问者访问其它用户的站、登录其它用户站（未登录主站）后访问自己的站、登录自己的站（未登录主站）后访问主站。

[0436] 以上所述的仅是本发明的优选实施方式，应当指出，对于本技术领域中的普通技术人员来说，在不脱离本发明核心技术特征的前提下，还可以做出若干改进和润饰，这些改进和润饰也应视为本发明的保护范围。

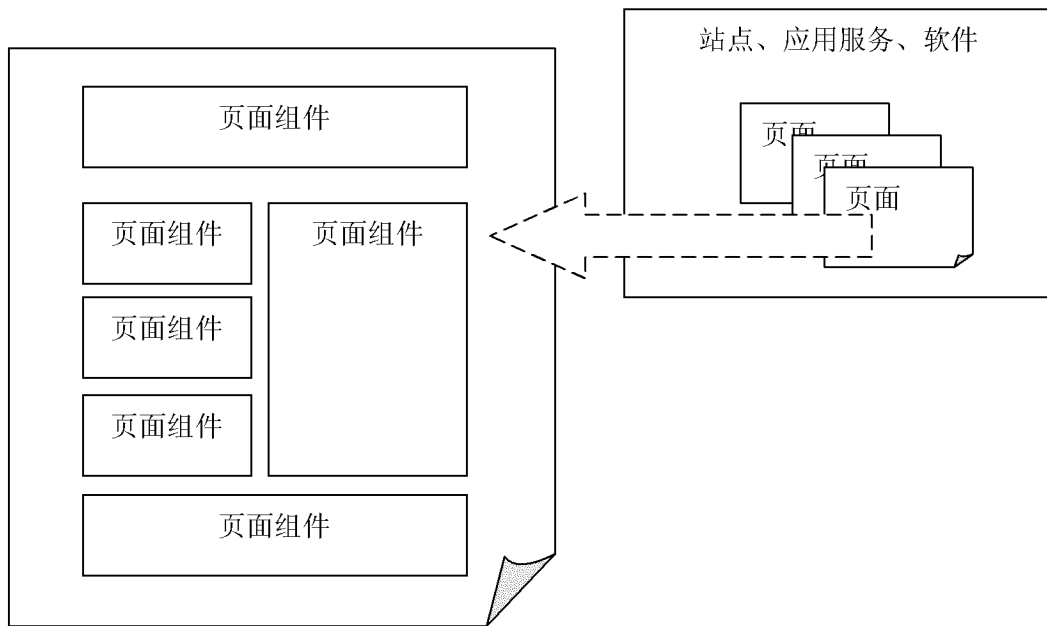


图 1

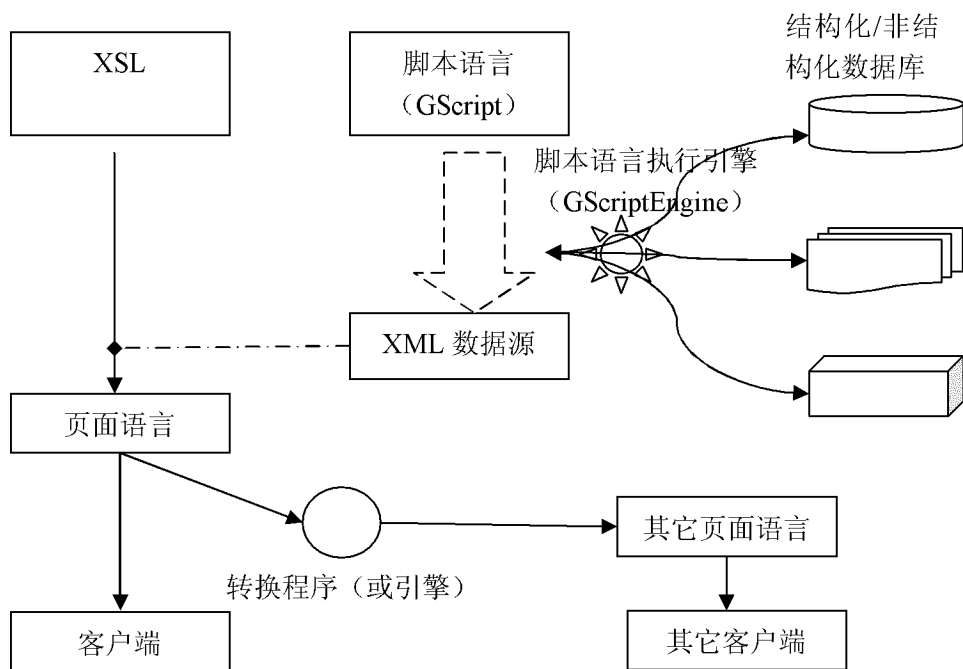


图 2

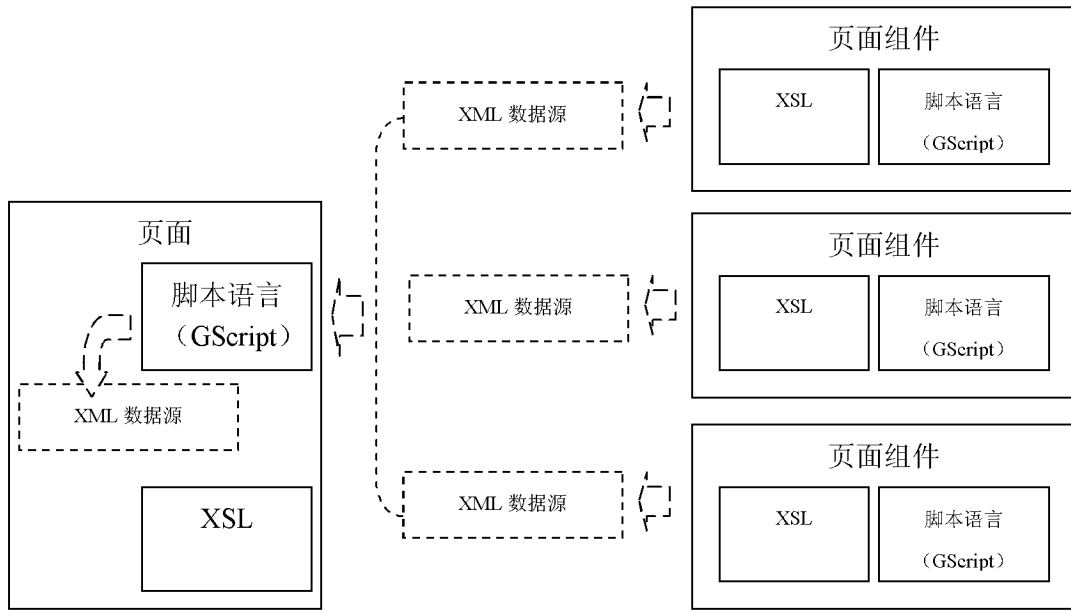


图 3

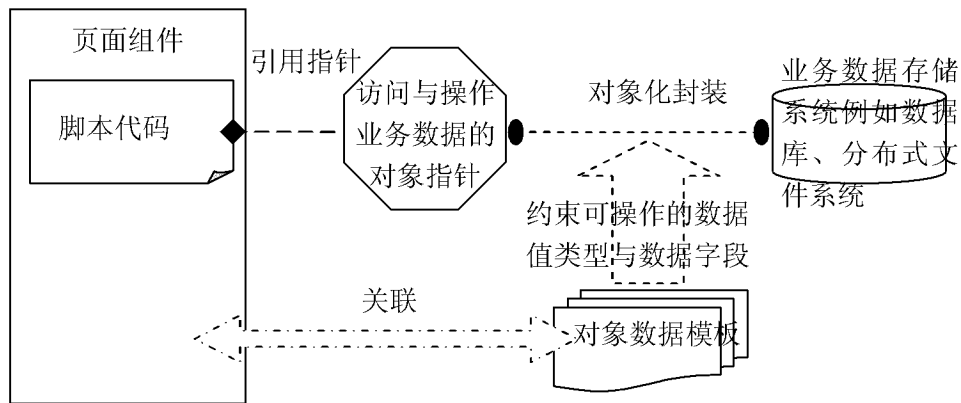


图 4

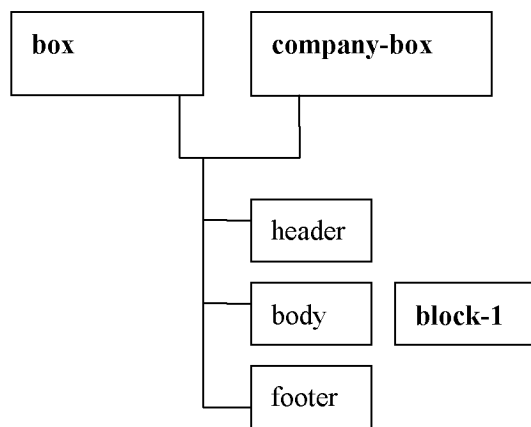


图 5

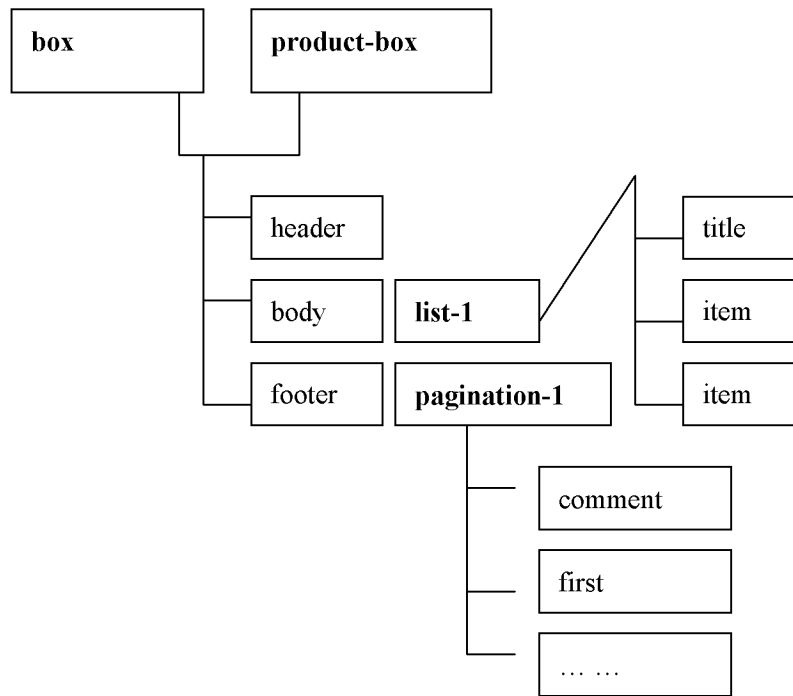


图 6

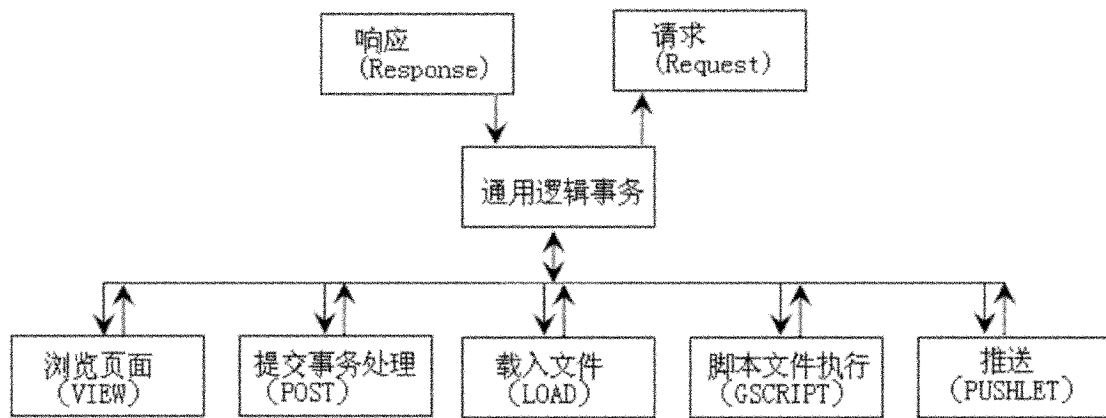


图 7

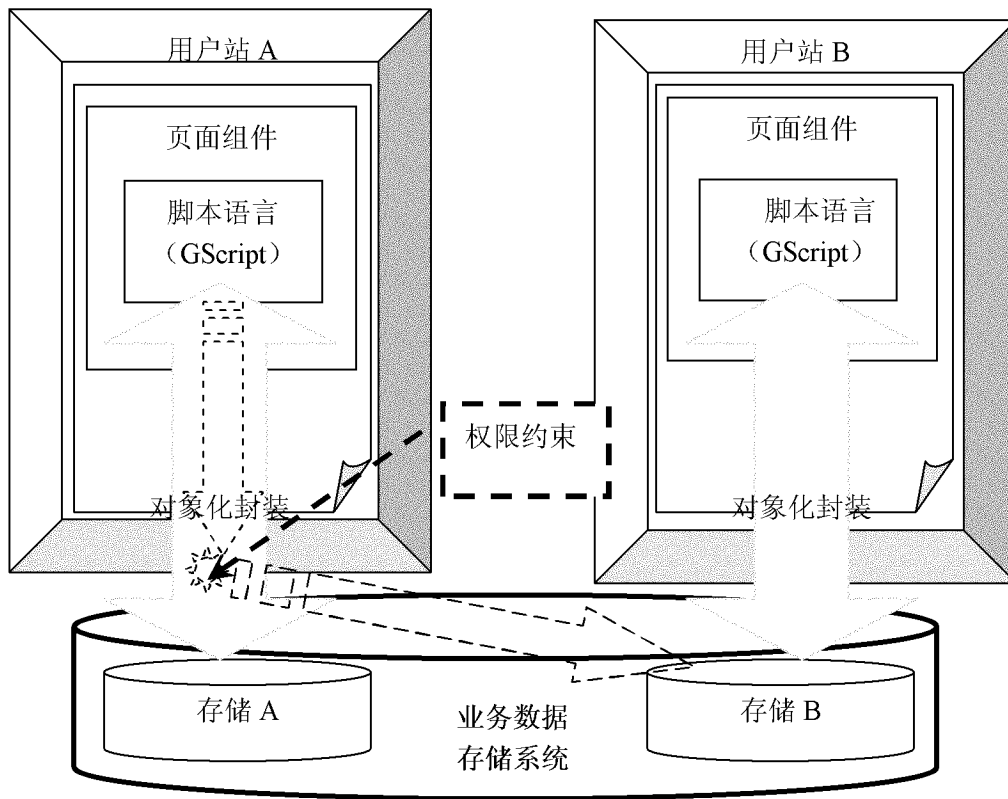


图 8

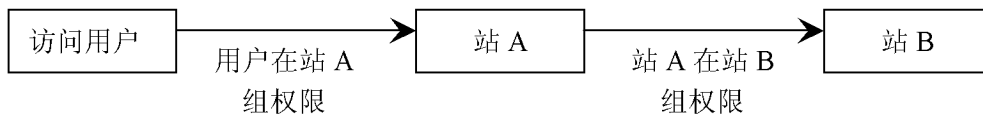


图 9

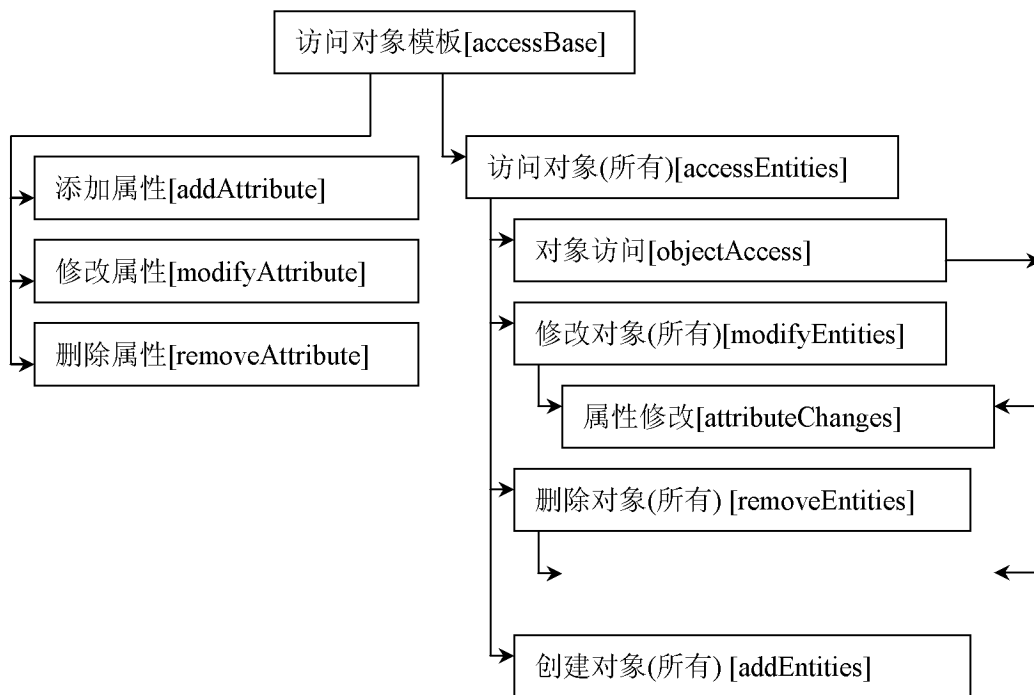


图 10



图 11

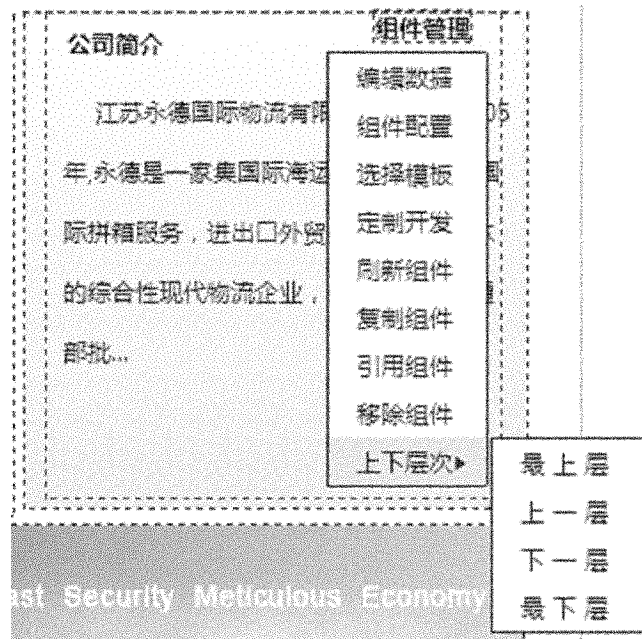


图 12

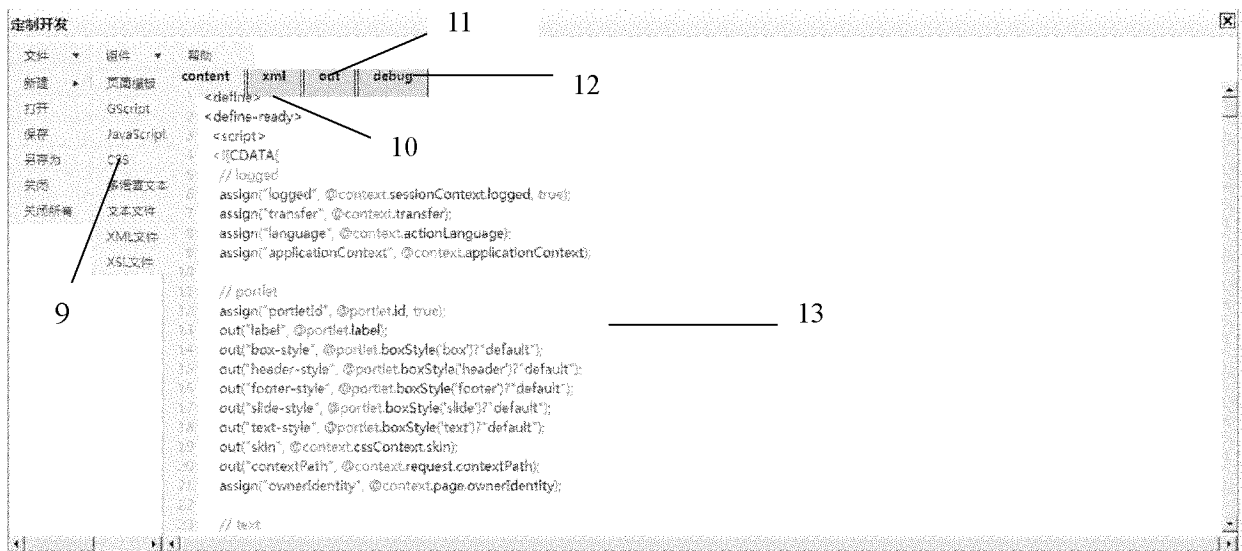


图 13

名称	别名	关键字	数据类型	关联对象	用户自定义	创建时间	创建者	与语言无关	输入安全	默认排序	备注
id	编号	是	字符串型(string)		否	2010-08-09 09:36:08.0			否		
createTime	创建时间	否	时间型(htime.timestamp)		否	2010-08-09 09:36:08.0			否		
creator	创建者	否	对象型(hthing.template#)	会员用户(user)	否	2010-08-09 09:36:08.0			否		
name	名称	否	字符串型(string)		否	2010-08-09 09:36:08.0		否	否		1
parent	父类型	否	对象型(hthing.template#)	新闻类型(newstype)	否	2010-08-09 09:36:08.0			否		root
remark	备注	否	字符串型(string)		否	2010-08-09 09:36:08.0		否	否		

添加 编辑 删除

图 14

编号	创建时间	创建者	名称	父类型	备注
1CIT40G18rab5	2010-08-10 10:21:23	yongdde	货运知识 [下级目录]	root	
T2MbeUo16i2as	2010-08-10 10:09:28	yongdde	航务公告 [下级目录]	root	
L281r21C8C015	2010-08-09 10:34:42	yongdde	行业新闻 [下级目录]	root	
1C113C1C11j68	2010-08-09 10:33:31	yongdde	公司新闻 [下级目录]	root	

添加 编辑 删除 4条记录 <<< 1/1 >>> [GO]

图 15

经营模式 单击这里选择 清除

- 生产加工
- 经销批发
- 招商代理
- 商业服务
- 其它模式

图 16

员工人数 单击这里选择 清除

- 5 人以下
- 5 - 10 人
- 11 - 50 人
- 51 - 100 人
- 101 - 200 人
- 201 - 300 人
- 301 - 500 人
- 501 - 1000 人
- 1000 人以上

图 17

主营行业

<input checked="" type="radio"/> 照明工业	<input checked="" type="radio"/> 车灯	<input type="checkbox"/> 大灯、前照灯
<input type="radio"/> 电工电气	<input type="radio"/> 手电筒	<input type="checkbox"/> 转向灯、角灯
<input type="radio"/> 传媒、广电	<input type="radio"/> 白炽灯	<input type="checkbox"/> 尾灯
<input type="radio"/> 安全、防护	<input type="radio"/> 照明器材代理加盟	<input type="checkbox"/> 雾灯
<input type="radio"/> 包装	<input type="radio"/> 插座	<input type="checkbox"/> 仪表灯
<input type="radio"/> 纸业	<input type="radio"/> 电子产品包装	<input type="checkbox"/> 刹车灯
<input type="radio"/> 办公、文教	<input type="radio"/> 专门用途灯具	<input type="checkbox"/> 车顶灯
<input type="radio"/> 数码、电脑	<input type="radio"/> 室外照明灯具	<input type="checkbox"/> 工作灯、检修灯
<input type="radio"/> 电子元器件	<input type="radio"/> 室内照明灯具	<input type="checkbox"/> 其他车灯

图 18

地区

<input type="radio"/> 江苏	<input type="radio"/> 抚州
<input checked="" type="radio"/> 江西	<input type="radio"/> 赣州
<input type="radio"/> 辽宁	<input type="radio"/> 吉安
<input type="radio"/> 青海	<input type="radio"/> 景德镇
<input type="radio"/> 山东	<input type="radio"/> 九江
<input type="radio"/> 山西	<input type="radio"/> 南昌
<input type="radio"/> 陕西	<input type="radio"/> 萍乡
<input type="radio"/> 四川	<input type="radio"/> 上饶
<input type="radio"/> 新疆	<input type="radio"/> 新余

图 19



图 20



图 21

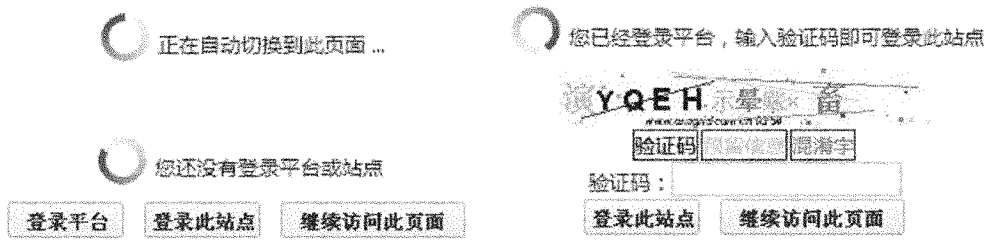


图 22



图 23