

可视化 Web 设计器^①

刘一田, 刘士进

(国网电力科学研究院, 南京 210037)

摘要: 描述了一种可视化 Web 设计器 VWD(Visual Web Designer)的设计思想、体系结构和实现技术, VWD 包括 5 部分: 基于 IFDL 的界面交互描述语言、模式抽取方法、基于浏览器的界面原型设计器、基于 Eclipse 插件的可视化 Web 设计器、编译引擎等。界面交互描述语言定义了可复用的 Web 结构, 模式抽取方法给出了模式定义及抽取方式, VWD 界面原型设计器加速了应用需求快速确认, Eclipse 插件版设计器支持与数据模型服务的对接和调试, 方便协作式前后端分离开发。应用实例及评估表明, VWD 提高了应用开发效率。

关键词: 可视化 Web 设计器; IFDL; 模式抽取方法

Visual Web Designer

LIU Yi-Tian, LIU Shi-Jin

(State Grid Electric Power Research Institute, Nanjing 210037, China)

Abstract: This paper describes design ideas, architecture and implementation techniques of a kind of visual Web designer VWD (Visual Web Designer). VWD consists of five parts: interactive interface description language based on IFDL, a pattern extraction method, a interface prototype designer based on browser, a visual Web design based on Eclipse plug-in, a compiler engine. Interactive interface description language defines reusable Web structure. Pattern extraction methods describe the schema definition and extraction mode. VWD accelerated interface prototyping application's needs to be quickly identified. Eclipse plug-in based designer facilitates docking and debugging in data models services facilitates collaborative development by separating frontend from backend. Application examples and assessments indicate that VWD improved application development efficiency.

Key words: visual Web designer; IFDL; pattern extraction methods

1 引言

随着 Web 应用系统所处的环境越来越开放和难控, 构成系统的组件也越来越异构和复杂, 使得系统的维护与演化的成本不断增加。现有的软件开发方法并不能很好地满足 Web 应用软件的设计需求, 逐渐显示出实用性和交互性的限制。如何在开放和动态环境下快速响应需求, 构造柔性架构的 Web 应用, 有效降低应用开发复杂性成为了新的技术挑战。

可视化 WEB 设计器提供了良好的解决方案。基于可复用组件库的可视化界面建模、代码生成及实时预览的快速开发工具, 是提高 Web 应用系统开发效率与产品质量的一种重要手段, 其关键技术是 Web 组件复用方法和可视化建模设计方法。实现 Web 组件复用

的主要方法是 Web 系统架构与 Web 组件开发技术, 通过 Web 系统架构设计把系统各模块组件化, 将 Web 组件组装成一个完整的 Web 应用系统。传统的可视化设计器侧重于仿真界面原型设计, 适用于用户需求确认, 对应用开发工作帮助有限, 特别是在应用开发的前后端联调测试时实用化程度较弱。

本文设计的可视化 WEB 设计器分离了界面原型设计与实际界面设计开发的过程, 其中基于浏览器的界面原型设计器支持原型设计和本地模拟数据, 用作需求分析; Eclipse 插件版设计器支持与真实业务模型服务对接, 方便开发人员调试和集成。从四个方面对 Web 应用的柔性架构及界面交互体验提供了支持: (1) 借鉴标准化界面交互流程模型, 提高 Web 应用标准化

^① 收稿时间:2015-01-28;收到修改稿时间:2015-04-02

水平; (2)离线设计支持和面向资源的 REST 服务支持; (3)支持单页应用程序(SPA)设计; (4)模式抽取功能, 抽取的典型场景可作为建模知识和界面模式库复用。

2 相关工作

Web 组件复用在 Java EE 中有标准规范, JSF 是用于构建 Java Web 应用程序的标准框架, 提供了以组件为中心的用户界面构建方法, 简化了应用程序开发, 但基于服务端事件驱动的 JSF 把事件处理模型限制到了服务器上, 制约了响应性及交互设计。

可视化 Web 建模设计的关键方法是使用标准化的 Web 交互语言对 Web 界面交互建模过程进行描述, 对 Web 系统架构进行建模, 目前国内缺少系统化的 Web 应用建模方法, 米兰理工大学的 Marco Brambilla 等阐述了交互流程模型语言 IFML^[1](Interaction Flow Modeling Language)定义, 以及已实用化的开发平台 WebRatio。IFML 已被国际标准化组织 OMG 采纳, IFML 的设计避免了表达交互设计时的图形和概念冗余, 提供了交互设计定义的可扩展性(例如, 新的界面组件或事件类型定义), 确保了模型级复用等。IFML 由 Web 建模语言 WebML^[2]演化而来。IFML 定义的主要建模概念包含视图容器、视图组件、事件、活动、提供数据传输的导航流和数据流、参数绑定、约束、模块化和上下文感知等概念。其中视图容器可包含视图组件, 视图容器和视图组件之间通过事件进行通信, 响应用户交互或系统默认的事件, 事件同时触发的活动执行后会更新界面交互状态, 并通过关联了数据流或导航流的参数绑定机制指定界面元素之间的输入输出依赖。基于上述已定义概念可灵活扩展个性化的界面交互描述元素, 形成统一的前端接口组成、用户交互、事件管理的标准规范, 将业务逻辑、数据模型和界面展示串联起来, 使得界面交互设计者和开发者之间关注点清晰分离, 增强 UI 设计者和组件开发者的沟通。

Web2.0 应用中 SPA(Single Page Application)趋于主流。SPA 一般采用主流的 MV*架构, 传统的 MVC 架构主要通过 Struts 实现, 其中视图负责前端的渲染, 是通过服务端输出 HTML 的渲染, 在 SPA 时代, 浏览器端形成了独立的 MVC 等层次, 可以较好地适用于基于可视化 WEB 设计器的设计和开发, 本文设计的可视化 Web 设计器生成的是基于 FWF^[3]框架的 SPA 应用代码。

3 可视化Web 设计器框架设计

可视化 Web 设计器 VWD 是一个面向企业信息管理系统的用户界面设计与开发工具, 主题交互结构分为基于浏览器的界面原型设计器和 Eclipse 插件版设计器两部分, 浏览器版本提供离线设计的需求分析功能, 数据采用本地对象化的 JSON 模拟数据, Eclipse 插件版提供应用开发和集成功能, 从图 1(a)、图 1(b)可以看出 VWD 的工作流程如下:

① 需求分析人员根据业务需求对用户应用领域的交互模型建模, 参考或导入已建系统的建模知识库和界面模式库, 来确定整个交互系统的风格和模式。

② 采用浏览器版的可视化设计器进行界面建模。根据交互视图的需求风格和模式, 拖拽设计器中的布局组件到设计器的画布中, 然后拖拽界面交互组件到布局组件的各个面板内。例如, 左树右表的界面模式, 拖拽垂直分割布局的布局组件→拖拽数据树组件到左侧→拖拽数据表格组件到右侧。界面设计器大纲视图中以分层树展示设计的界面结构等内容。

③ 界面交互数据绑定。在界面设计器数据模型管理工具中定制客户端本地 JSON 数据模型, 然后在画布中选择界面交互组件, 匹配要管理的本地数据模型。

④ 编写视图模型中的基本界面交互逻辑。根据业务应用需求, 确定界面交互组件之间的数据流传递方向和绑定参数, 编写界面交互组件触发的导航、增删查改等事件, 在事件处理函数中处理数据流以及相关的参数绑定。上述定义确定了界面交互描述文件 UID, UID 是一个 zip 文件, 包含了视图、视图模型相关的脚本文件和界面描述文件, 界面描述文件使用 IFML 进行描述, 并根据需要扩展了 IFML 定义, 增加了 Weblet 依赖、权限对象、视图模型以及界面交互逻辑脚本等资源文件位置的定义。

⑤ 预览原型。运行时调用编译服务生成前端 FWF^[3]框架代码, 启动前端框架渲染引擎渲染主视图, 和用户确认功能需求是否满足, 根据用户意见修改原型。原型确定后使用模式抽取工具抽取典型界面原型。

⑥ 原型确定后, 提交界面交互描述文件给应用开发组, 应用开发人员在 Eclipse 插件版可视化设计器中导入原型描述文件 UID, 插件版界面设计器解析界面交互描述文件生成插件版界面视图, 以及符合 FWF 框架的 Weblet 结构^[3], 其中包含了界面交互逻辑脚本文件。

⑦ 真实数据模型绑定和调试. 选择插件版面布中界面交互组件, 在数据模型管理器中选择 REST 服务, Eclipse 方便 REST 服务的集成和调试, 而浏览器版可视化 Web 设计器无法较好支持 REST 服务调试功能.

⑧ 根据需求调整交互逻辑脚本, 确定界面视图和视图模型的业务交互逻辑.

⑨ 编译插件版定义的界面定义语言, 生成最终可部署运行的应用代码.

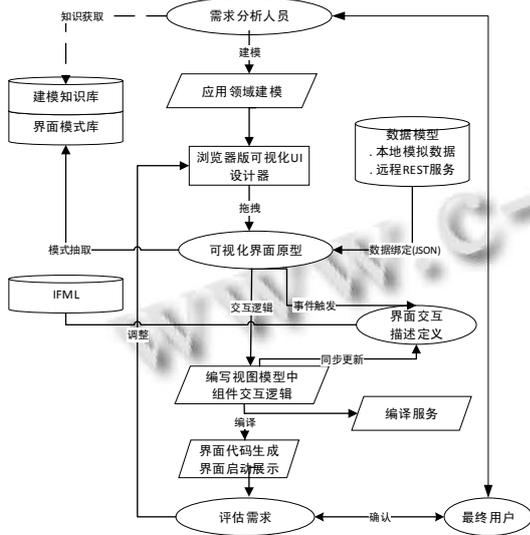
4 可视化Web设计器的实现方法

可视化 Web 设计器 VWD 采用组件复用和可视化拖拽技术, 交互界面采用拖拽生成方式, 将视图中通用的交互逻辑方法进行提炼, 形成建模知识库和界面模式库, 通过编译引擎生成代码, 可实时预览已有设计结果, 方便根据需求快速调整设计. 根据上述 VWD 的框架结构的设计, VWD 包含几点主要关键技术: ① 基于 IFML 的界面描述语言扩展定义, 关键概念定义见表 1; ② 基于描述语言的编译引擎; ③ 模式抽取方法; ④ 可视化设计和编辑器技术. 关键技术分别描述如下.

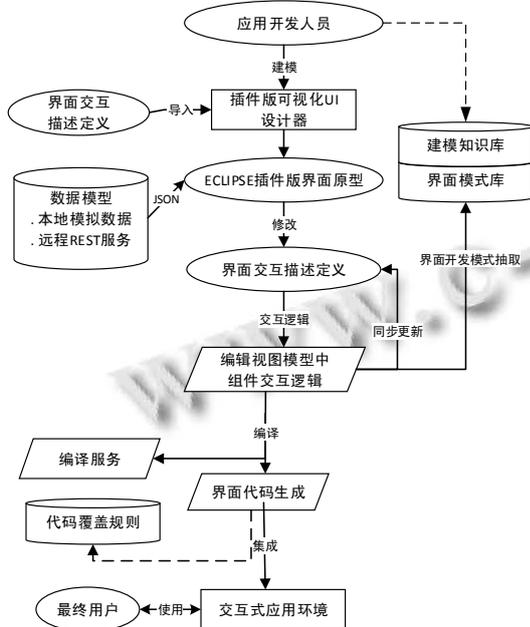
(a) 基于 IFML 的界面描述语言扩展定义

表 1 VWD 基于 IFML 的扩展定义

概念	含义	实现示例
应用小场景	可复用的交互应用场景组件.	Weblet
视图容器	包含显示内容和支持交互的接口元素.	窗体、面板
嵌套视图容器	包含子视图容器的视图容器, 子视图容器可以交替显示.	选项卡
视图组件	显示内容并接受输入的元素.	表格、表单、编辑器.
事件	改变应用状态发生的一个时刻.	创建数据前、创建数据后
动作	事件触发的一系列业务逻辑.	数据校验、更新
导航流	输入输出依赖.	HTTP 请求发送和接受
数据流	用户交互触发的视图组件和动作之间的数据传递.	
参数	类型化的指定值.	HTTP 查询参数, HTTP POST 参数
参数绑定	输入参数和输出参数关联的规范.	
输入端口	模块和其环境接受输入参数的交互点.	
输出端口	模块和其环境输出参数的交互点.	
视图组成部分	不能单独存在的视图组件, 可以触发事件、接受输入, 仅能作为视图组件的一部分存在.	表单域
视图控制器	视图中组件加载和界面交互逻辑定义	
权限对象	角色对资源的控制力度	可见性、可操作性定义



(a) VWD 的原型设计器结构



(b) VWD 的Eclipse插件设计器结构

图 1 设计器结构

(b) 基于界面描述语言的编译引擎

根据(a)中界面交互建模语言定义生成 XML 格式的界面描述文件, 并将界面交互逻辑脚本文件同时打包

到一个 zip 格式的压缩文件中, 重命名文件后缀为 ".ui", 编译引擎将 ".ui" 文件作为输入, 根据转换映射规则, 生成 SPA 模式的 Weblet 代码, 编译过程如图 2 所示。同时, ".ui" 文件作为可视化 Web 设计器 VWD 的输入, 将可视化设计场景的设计界面进行还原以重构原型设计状态。

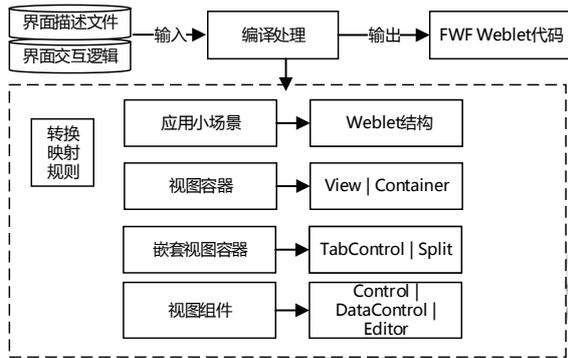


图 2 编译引擎处理过程

(c) 模式抽取方法

可复用的界面模型是一个四元组^[4], 可描述为 $RUI = \langle DM, PM, UM, IM \rangle$, 其中 DM 为领域模型, PM 是展现模型, UM 是用户模型, IM 是交互模型。

定义 1. 领域模型 DM 定义为 $DM = \langle \{Entity\}, \{ER\} \rangle$, 其中 $Entity = \{Attr_1, Attr_2, \dots, Attr_n\}$, 实体表示含有若干属性的业务对象, ER 表示实体之间的属性引用关系。

定义 2. 展现模型 PM 定义为 $PM = \langle \{VCC\} \rangle$, 其中 VCC 表示 1 或多个可视化组件组合, $VCC = \langle \{DP, VP, IP\} \rangle$, DP 表示数据属性, VP 表示可视化属性, IP 表示交互属性。

定义 3. 用户模型 UM 定义为 $UM = \langle User, Role, Permissions \rangle$, User 代表用户对象, Role 表示用户角色, Permissions 主要定义用户的角色及对组件可见性和可操作性的权限。

定义 4. 交互模型 IM 定义为 $IM = \langle \{CIE\}, \{CIEA\} \rangle$, CIE 表示组件交互事件, CIEA 表示事件监听器, 描述了组件交互事件引发的一系列业务逻辑动作^[5]。

模式抽取的过程即根据 RUI 的定义对已定义的界面模型元素进行归约的过程, 归约过程如下:

① 根据领域知识建立领域模型, 确定领域模型所覆盖的实体对象, 分析各实体对象的属性, 然后将若

干相关属性组织起来, 定义为实体(Entity), 并把一个实体属性来源于另一个实体的属性定义为关联(ER), 根据业务领域知识进行命名, 直到所有的领域信息用实体联系模型描述完成。

② 设计展现模型需要明确界面中的可视组件对象, 包括交互对象和应用对象, 应用对象是用户交互的场景, 交互对象是可视化组件。

③ 确定用户模型, 设计不同角色对展现模型中的应用对象和交互对象的资源访问和增删查改、可见性等操作权限。

④ 交互模型的构建主要是对交互上下文环境进行描述: 包括交互事件、应用反馈等。具体来说, 是描述在用户界面中要执行的原子操作(如单击、双击等); 结合上下文, 描述出对应展现模型中交互对象的命令(如激活、更新、删除); 根据交互模型触发数据流和导航流, 完成交互过程。

(d) 可视化设计技术

VWD 原型设计工具基于柔性框架 FWF 提供的客户端组件实现, Eclipse 版可视化设计工具基于插件图形化编辑框架 GEF(Graphical Editor Framework)和建模框架 EMF(Eclipse Modelling Framework)实现^[6], GEF 中的命令封装了对模型的修改, 通过继承 GEF 中的抽象类命令, 提供 Redo/Undo 功能, 通过 Xml 数据绑定机制实现已设计界面的还原功能。

可视化拖拽设计的关键技术点包含画布布局、鼠标位置捕获和提示、工具箱可视化组件图元、属性设置、大纲视图、上下文菜单等。画布布局确定了画布中可视化组件图元元素的位移定位方法; 鼠标位置捕获根据鼠标距离左上角的偏移量、可视化组件之间的逻辑关系以及鼠标所在布局容器中的边界, 确定图元元素在画布中的实际位置并给出是否允许拖拽的提示; 属性设置提供了可视化图元组件属性的键值对设置; 可视化组件图元的上下文菜单提供了常用的复制、粘贴、撤销、重做等功能, 数据组件提供了额外的匹配数据源的功能; 大纲视图根据界面描述文件以分层导航树的形式提供了已设计界面元素的导航功能, 并提供了外部数据源的依赖注入功能。

5 应用实例

可视化 WEB 设计器 VWD 已在国家电网公司统一应用平台(SG-UAP)中实现并大规模应用, 这里以国

国家电网公司车辆管理系统的一个应用场景为例,介绍使用 VWD 的应用开发过程。

参照第三节中 VWD 的工作流程,用浏览器版可视化 Web 原型设计工具确认界面原型,生成界面交互描述文件,将界面交互描述文件导入 Eclipse 版可视化 Web 设计器,还原界面设计模型,根据模式抽取方法提炼出车辆管理领域模型、展现模型、用户模型、交互模型。在 Eclipse 中首先完成领域模型的服务开发和模型注入,然后将用户模型配置关联到可视化数据组件中,确定不同角色的可访问性和可操作性,最后将交互模型重构后注入到事件函数中,同时在合适的交互事件中加入业务逻辑处理脚本代码,完成上述操作后的界面如图 3 所示,调用编译引擎生成完整的应用客户端代码,最后进行预览、设计调整和代码交付工作。



图3 Eclipse版设计器的设计开发视图

6 效率评估

给定一个应用场景,从应用开发组中抽调平时代码质量高的八名开发人员,分为 A、B 两组,对 A 组四位人员培训可视化 WEB 设计器的使用, B 组四名人员强化基于代码的应用开发,在需求不确定的前提下, A 组首先完成了需求的确认,并提前 1 天完成了应用场

景开发。在需求已确定的前提下, A 组直接使用 Eclipse 版可视化设计器完成了界面的设计,提前半天完成应用场景的开发。

目前,国家电网公司设备(资产)精益运维管理系统等大规模系统的应用实践表明,可视化 WEB 设计器 VWD 对提升开发效率效果明显,节约了研发成本。

7 结束语

本文研究了基于 IFML 的界面描述语言和模式抽取方式,在此基础上,给出了界面描述语言扩展定义和界面模式抽取方法,提出了一个需求原型设计和应用开发分离的可视化 WEB 设计器设计方法,阐述了该设计器的实现架构以及关键实现技术。最后,以基于国家电网公司统一应用平台的应用开发实例为背景,给出一个快速界面原型和应用开发的实现过程。应用实例、效率评估及开发实践表明, VWD 降低了 Web 应用开发的学习成本,提高了 Web 应用开发效率。

参考文献

- 1 Object Management Group. IFML. <http://www.omg.org/spec/IFML/1.0/Beta2/>.
- 2 陈勇涛.基于 WebML 的 Web 应用程序建模研究[硕士学位论文].北京:对外经济贸易大学,2007.
- 3 刘一田,刘士进.一种柔性 Web 展现框架模型.计算机系统应用,2013,22(12):155-159
- 4 吴刚,董金祥,何志均,陈刚.可视化、智能化用户界面管理系统的模型及实现.软件学报,1997,8(4):283-290
- 5 曹元大,韩雪来.跨平台可视化网页设计器的设计与实现.北京理工大学学报,2001,21(1):105-108
- 6 谭凯.基于 ECLIPSE 的 WEB 应用可视化组件管理平台的研究与实现[硕士学位论文].北京:北京工业大学,2009.