

Mashup 组件技术研究*

刘芳^{1,2} 李春旺¹ 王昉¹

1 (中国科学院国家科学图书馆 北京 100190)

2 (中国科学院研究生院 北京 100190)

[摘要] 文章在介绍 Widget 的概念、类型、特点基础上,重点分析了 UWA、W3C Widget 对象结构、API 标准规范以及基于 Widget 的集成融汇应用平台架构,讨论嵌入式 Widget 应用形式、实现方法、存在的问题及未来发展方向。

[关键词] 集成融汇 Widget 体系架构 表示层应用

[分类号] G250.7

Study on Widgets for Mashup

Liu Fang^{1,2} Li Chunwang¹ Wang Fang¹

1(National Science Library, Chinese Academy of Sciences Beijing 100190, China)

2(Graduate University of Chinese Academy of Sciences, Beijing 100190, China)

[Abstract] After introducing the definition, type and feature of widgets, this paper emphatically analyzes object structures and API standards of UWA and W3C widgets, and discusses the architecture of enterprise mashup and implementations of the mashup application based on widgets. Finally, related work and trends are given.

[Keywords] Mashup Widget Architecture Presentation application

1 Mashup 组件

Mashup 组件 (Widget) 是一种小的客户端交互应用系统,用于显示、更新本地数据或网络数据,它按一定规范进行封装,支持在用户端计算机或移动设备上的下载、安装,它既可以作为一个独立的应用程序运行在 Web 浏览器之外,也可以被方便地嵌入到一个网页里^[1]。从应用角度上可以将 Widget 分为三种形式:第一种是桌面 Widget (Desktop Widget), 又称 Software Widget, 是一种可以在 Windows XP/Vista、Linux、Mac OS X 等操作系统上独立运行的小程序^[2],如 Yahoo!桌面 Widget、Windows Vista 左边栏组件 gadget、Apple Dashboard 等。第二种是网页 Widget (Web Widget), 又被称为 gadget、badge、module、webjit、capsule、snippet、flake 等,它起源于代码复用思想,表现为一组可移植的 HTML+JavaScript 代码片段,一次编写无需任何编译就可以被用户集成到任意基于 HTML 格式的网页里^[3]。网页 Widget 只有服务功能定义部分,运行环境与 Widget 分离,最常用的运行环境就是通用浏览器。第三种是移动 Widget (Mobile Widget),它与桌面 Widget 类似,只是运行在移动终端设备及专用移动操作系统上,移动 Widget 支持基于位置感知的动态 Web 应用构建,支持用户动态选择最相关信息,避免冗余的数据传输带来的额外流量^[4]。

Widget 技术发展促进了 Web 集成融汇的创新。集成融汇服务主要包括数据融汇、功能融汇、表示对象融汇三种类型,其中,表示对象融汇即指 Web Widget 融汇。一方面,利用

* 本文系中国科学院2009年新增能力项目“知识服务集成融汇关键技术研究”成果之一。

Widget 技术，图书馆可以将网站内容封装成一些比较小的服务单元，支持任意第三方组织机构的集成调用，形成服务传播的长尾效应。另一方面，科研用户通过对 Widget 的融汇组合，可以方便、快捷地构建个性化应用系统等，从而使用户成为网络服务的构建者。

2 Widget 架构

目前存在多种 Widget 标准规范，比较有影响的包括 Netvibes 的 UWA、Google Gadget、Yahoo Widget、Opera Widget^[5]、Intel Widget 等。不同的 Widget 从结构、功能到操作方法上存在差异。如 Opera Widget 支持第三方插件和 JAVA 程序，对安全性进行了限定并且提供了四种可选择的 Widget 模式；Google Gadget 由一个 XML 文件定义，在<CDATA>中嵌入 HTML、JavaScript 代码；Intel Widget 则要求输入数据要具有统一的格式。本节以 Netvibes UWA、W3C Widget 为例，介绍 Widget 体系结构以及跨平台操作规范。

2.1 UWA

UWA (Universal Widget API) 是 Netvibes 发布的一个 Widget 架构规范，它采用 XHTML 规范进行静态描述，界面风格采用 CSS，行为控制以及与 DOM 的交互操作功能采用嵌入式 JavaScript/Ajax 实现。UWA 是目前跨平台性最好的 Widget 标准规范：它不仅可以作为 Web Widget 支持 Netvibes、iGoogle、Opera、Yahoo!、Apple Dashboard 等多种主流用户桌面系统的集成；而且通过集成由 JavaScript 文件和 CSS 文件组成的仿真运行环境，UWA 可以输出为 Desktop Widget、Mobile Widget，支持在 Windows Vista、Mac OS X 等操作系统上的独立运行，支持移动设备的调用。

(1) UWA Widget 文档结构^[6]

一个 UWA Widget 文档必须采用标准的 XML/XHTML 编码格式，必须采用 UTF-8 保存格式，同时，为了支持 Widget 的平台兼容性，Netvibes 要求：一个 Widget 对象的所有编码信息都包含在一个静态的 XHTML 文件中，对外部数据的获取采取 Ajax 异步调用方法，数据解析处理以及交互显示控制等采用内嵌 Javascript 脚本语言实现。所有 Javascript 代码、CSS 代码都直接嵌入到 XHTML 文件中，唯一可以外部调用的是运行环境相关的 Javascript、CSS 文件。

一个标准的 UWA Widget 文档由三部分组成：XHTML 文件头、头标区<head>、正文区<body>。其中，XHTML 文件头除包括 XHTML 1.0 规范要求的必备内容外，还必须在<html>元素中包含 Netvibes Widget 命名空间属性：xmlns:widget=“http://www.netvibes.com/ns/”。正文区<body>用于定义 Widget 组件的基本功能以及交互界面，其定义方法和一个 html 页面一样。UWA 头标区<head>除包含 HTML 标准内容外，还包含以下特有内容：

第一，元数据描述信息，Netvibes 定义了一组元数据用于描述 Widget 对象及其工作方式，如：author、description、thumbnail 等，该内容不是必备项，但如果定义了相关元数据，系统在对 UWA Widget 对象进行操作时，会自动读取元数据信息并进行交互反馈。

第二，仿真运行环境信息，Netvibes 提供了 Widget 独立运行环境文件，包括界面风格定义 CSS 文件和交互控制 Javascript 文件。为支持 Widget 以桌面 Widget 方式在操作系统上独立运行，需要在 XHTML 文件中包含两个仿真运行环境文件：

```
<link rel="stylesheet" type="text/css"
```

```
href="http://www.netvibes.com/themes/uwa/style.css" />
<script type="text/javascript"
src="http://www.netvibes.com/js/UWA/load.js.php?env=Standalone"></script>
```

第三，参数信息<widget:preference>，UWA 支持多种类型的数据参数、属性参数定义，支持用户对 Widget 可编辑部分进行参数设置。

(2) Widget API^[7]

为支持 Widget 开发，Netvibe 提供一个基于 Javascript 框架的 Widget 对象类及 API，封装了 Widget 属性、事件和方法，支持对 Widget 的使用和操作、数据请求和交换、参数调用及样式和元素的扩展等。由于 Netvibes 运行环境会自动建立 Widget 对象实例，开发人员利用 Javascript 代码可以直接调用 Widget 事件、方法。例如：调用 widget.onResize 获取 Widget 大小调整事件，调用 widget.getValue(name)读取<widget:preference>下指定元数据信息，调用 widget.setTitle(title)设置 Widget 标题，UWA.Data.getJson(url, callback)则从指定数据源返回一个 JSON 格式的数据。

UWA 良好的兼容性使其受到广泛关注。SOHU 公司基于 UWA 推出自己的 SOW(SOHU Open Widget)。为推动 Widget 应用发展，Netvibes 创建了基于 Widget 的集成融汇生态系统(ecosystem)，鼓励用户参与 Widget 云的建设，并提供开放的调用服务，目前已经收集到 17.5 万个 Widget，每月提供的 Widget 调用服务超过 5 亿次^[8]。

2.2 W3C Widget

为解决多种Widget规范并存所造成的跨平台兼容问题，更好地支持Widget在用户桌面、移动设备、多语言环境中的兼容调用，W3C积极推动Widget标准规范建设。下图是W3C提出的Widget技术结构图：

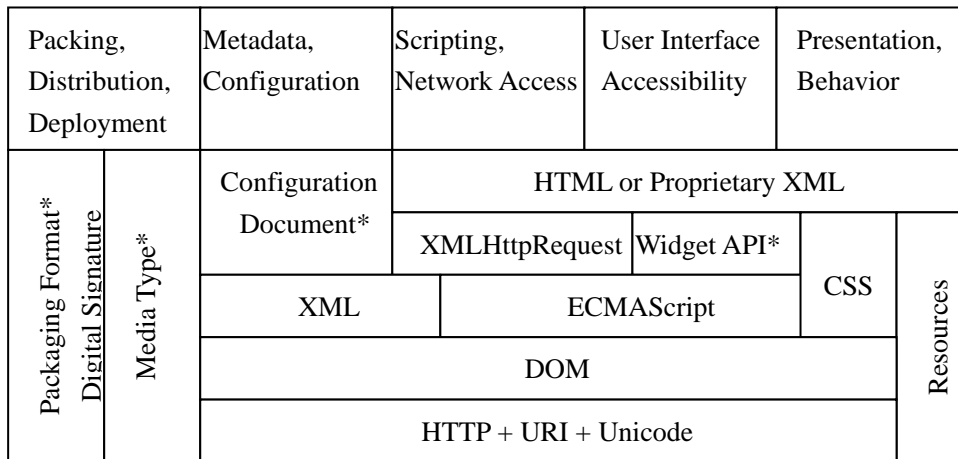


图1 Widget技术结构图^[9]

从功能层面看，Widget包括分发包（Packaging）、元数据定义（Metadata）、基于脚本语言的功能定义（Scripting）、用户界面（User Interface）以及表现行为（Presentation）等部分。从技术层面看，Widget是基于HTML、CSS、ECMAScript、XML等技术规范编写的一种应用，它采用异步请求方法XMLHttpRequest访问外部数据，通过封装标准接口API实现Widget与运行环境（Widget User Agent）以及Widget之间的互操作。在以上Widget技术中，有些尚未形成统一遵守的标准规范（图中带星号内容），包括Widget接口API规范、对象包格式规范、媒

体类型规范以及配置文档规范等，它们构成了Widget标准化研究与建设的重点。

目前，W3C已经推出了Widget 1.0系列标准草案（Widgets 1.0 family of specifications）。首先，W3C推荐一个通用的Widget对象包规范（Widgets 1.0: Packaging and Configuration）^[10]。一个Widget对象包主要包括以下内容：一个开始文件，可以位于Widget包的根目录或子目录中，可被内容元素的src属性引用；一个配置文件（Manifest），必须位于Widget包的根目录，采用xml语言编写并且必须以config.xml命名，用于声明Widget元数据及配置参数，每个配置文件至少要声明一个Widget元素和一个内容元素，内容元素必须包含属性：src；零个或多个图标文件，用于作者标记自制Widget；零个或多个资源文件，包括文本、图片、用户设计的界面标记、样式表、可执行脚本及Widget中其他可能文件；零个或多个数字签名，位于Widget包的根目录，实现对Widget访问的安全控制。其次，W3C推出了Widget API标准规范草案（Widgets 1.0: APIs and Events）^[11]，定义统一规范以便将API封装在自包含（self-contained）Widget对象中，支持用户利用API：访问Widget元数据、参数、实例属性，捕获Widget事件（如本地化信息），访问用户设备上的相关服务、资源和应用（如移动设备上的照像机、短信服务等），调用Widget运行环境功能完成数据处理以及构建设备独立的用户交互呈现等。除此之外，W3C还发布了Widget相关的数据签名规范^[12]、资源更新规范^[13]、网络资源请求规范^[14]、URI规范^[15]等，以全面推动Widget标准规范建设。

3 Widget 应用

3.1 基于 Widget 的企业 Mashup 平台

在传统门户系统中，对Portlet的集成操作主要在服务器端完成，由portlet容器装在portlet对象，执行后生成一表示层HTML代码片段，通过HTTP协议传递给客户端浏览器，并以<div>对象方式显示到指定位置。与portlet不同，融汇组件Widget可以直接运行在浏览器中，由浏览器负责装载Widget对象，包括相关功能脚本代码、CSS信息、外部数据等，并通过调用DOM交互方法将运行结果更新到DOM对象中，最终完成结果呈现。既可以运行在服务器端也可以运行在浏览器端的特点，使得Widget融汇服务能更好地体现以用户为中心的特点，因而受到广泛关注，并成为企业2.0（Enterprise2.0）建设的核心技术。目前，在众多企业Widget应用平台中，SAP的EMAP具有一定代表性(见图2)。

EMAP是一个基于Widget的应用开发平台，支持知识工作者或领域专家从Widget仓储中选择相关Widget，经组合、布局等操作后构建新的应用系统。EMAP数据层包括两类数据：企业数据以及Widget数据，前者支持客户端对企业平台上传统数据的访问，后者指对仓储系统中Widget信息的调用。仓储中的Widget组件可以由企业内开发人员创建并注册保存，这类组件对象保存Widget全部信息，包括表示逻辑、应用逻辑、元数据等。也可以将第三方平台系统上的Widget作为插件引入到本地仓储中，如iGoogle Gadget、Netvibes Widget等，这类插件在仓储系统中只保存元数据信息并绑定URL，Widget运行时系统自动通过URL调用远程平台上的Widget服务。

EMAP客户端主要由四部分组成：应用管理器（Application Manager）负责提供Widget的安装、卸载、生命周期管理、应用情景管理以及事件交换管理等；布局管理器（Layout Manager）负责按照主题需要灵活排列Widget以及对Widget的定义和应用陈述；数据管理

器（Data Manager）提供透明的企业数据服务访问，以便将其与本地、第三方 Widget 融合；Widget 仓储管理器（Repository Manager）建立客户端与后台 Widget 仓储管理系统的 REST/HTML 连接，支持用户选择、调用相关 Widget 组件。

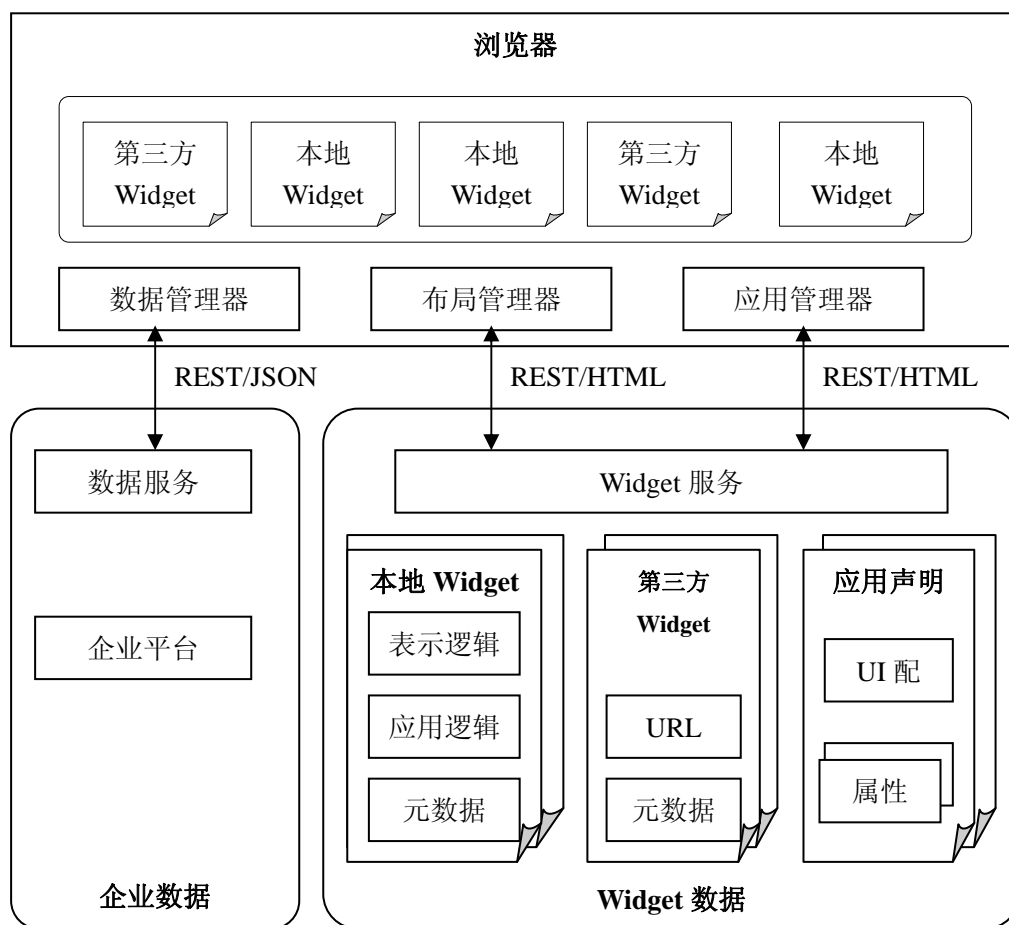


图 2 EMAP 企业 Mashup 平台^[16]

EMAP 平台体现了两个特点：一是对企业内部遗留系统中数据和服务的兼容，主要表现为数据层融汇和功能层融汇；二是建立新型的 Widget 融汇服务体系，支持表示层融汇，使得跨企业的服务集成更加简单、方便、快捷。这些特点对数字图书馆融汇服务平台建设具有一定借鉴性。

3.2 Widget 应用形式

Widget 支持多种应用形式。例如：国家科学图书馆（以下简称“国科图”）在参照国内外相关系统建设经验基础上，提出了自己的融汇服务平台建设思路，即在国科图 SRU 开放接口之上，构建 Widget 融汇仓储系统（图 3 为国科图融汇服务目录），封装、登记、保存图书馆基本服务组件群、专题服务组件群以及数字图书馆相关领域服务组件群，从而支持开放调用、动态组合、用户环境嵌入等多种应用。

(1) 组合多个 Widget 构建个人起始页面（startpage）。支持用户自我构建个人信息环境是 Web2.0 信息服务的重要特点。基于 Widget 技术、借助类似 EMAP 等平台工具，让即使没有计算机技术背景的科研用户也可以方便快捷地搭建个人主页、个性化信息空间，并且根据个人需求随时调整页面内容与布局，实现用户自我服务。国科图计划开发一个 iLibrary