

# 基于 WindowsCE 的嵌入式浏览器的设计

## Designing the Browser of an Embedded Software system which Based on WindowsCE

陈诗松 陈蜀宇 (重庆大学计算机系 重庆 400044)

**摘要:** 设计了一个基于 WinCE 平台的简单的嵌入式浏览器。该浏览器依据由 W3C 制定的 DOM 系列规范,在借鉴开源项目相关技术的基础上,以 Wince 为平台,研究了基于此平台的嵌入式浏览器的相关技术。其主要由网络连接模块、语法与语义分析模块、页面元素排版模块、图形图像显示模块等 4 个模块组成。能够完成基本的网页浏览任务。

**关键词:** DOM 嵌入式 浏览器 WINCE HTML

### 1 引言

嵌入式系统是以应用为中心,以计算机技术为基础的,软硬件可裁剪的,适应应用系统对功能、可靠性、成本、体积、功耗等各个方面的严格要求的专用的计算机系统。一般而言,嵌入式系统不是一个独立的计算机系统,而是作为其他系统的组成部分使用。

世界上第一个嵌入式系统诞生于 20 世纪 70 年代。到今天,嵌入式系统已经经过了大致 4 个发展阶段:以单芯片可编程控制器为核心的系统,同时配置检测、伺服、指示等功能;以嵌入式 CPU 为基础,以简单的操作系统软件为核心;以完善的嵌入式实时操作系统为标志的阶段,此一阶段的特点是操作系统内核精简、执行效率高、模块化且具备良好的可扩展性;以 Internet 为标志的嵌入式系统。

目前,嵌入式系统的发展已经进入了第四阶段,与 Internet 的结合。

近几年来,嵌入式系统市场快速发展,已经成为计算机领域的一个重要的组成部分,并成为了近年以来新兴的研究热点。目前,嵌入式产品的应用越来越广泛,其应用领域遍及航空航天、通信网络、信息家电、医疗教育、工业控制、金融管理等等。在人们的日常生活中,嵌入式系统也扮演着越来越重要的角色。

嵌入式浏览器作为网络应用中的一个重要组成部分,其作用是显而易见的。同时,由于嵌入式操作系统的多样性以及嵌入式浏览器与通用操作系统平台上的

浏览器的差异性,决定了嵌入式浏览器的应用范围是非常多样且千差万别的。就目前而言,还没有任何嵌入式浏览器可以满足所有嵌入式系统的要求,而达到象桌面操作系统上浏览器一般的垄断地位,在可预见的将来也尚无此种趋向。

### 2 Wince 嵌入式操作系统

Wince 操作系统是由 Microsoft 公司专门为嵌入式系统所开发的一个操作系统,此操作系统与一般的桌面操作系统的最大不同,就在于此操作系统是可定制的。

在使用 Wince 系统进行软件开发以前,必须定制一个适用的 Wince 系统,这也是由嵌入式系统本身的特点决定的。根据嵌入式系统所要运用的范围,切合实际的需要,选取 Wince 相关的组件,定制出一个适合所开发嵌入式系统的操作系统是十分必要的。

一般而言,定制 Wince 系统所使用的工具是 Platform Builder。使用此工具定制 Wince 系统需要经过以下几个步骤:根据所采用的硬件平台以及该系统的用途决定采用哪些组件;添加相关的组件,以及由硬件平台厂家所提供的 BSP 使用 Platform Builder 的相关命令生成 Wince 系统;使用 Platform Builder 的相关命令生成该 Wince 系统的 SDK;在 PC 机上安装该 SDK,并将之前生成的包含 Wince 系统的 NK 文件拷贝到嵌入式硬件平台上;使用 EVC 选择相应的 SDK 所包含的调试平

台,即可开始程序的调试与编制。

### 3 W3C 与 DOM 规范

#### 3.1 W3C

W3C(World Wide Consortium),又称为 W3C 理事会。该组织于 1994 年在麻省理工学院成立,其成立人正是万维网的发明者蒂姆·伯纳斯·李。建立此一组织的初衷在于解决 WEB 应用中,不同的开发者、平台、

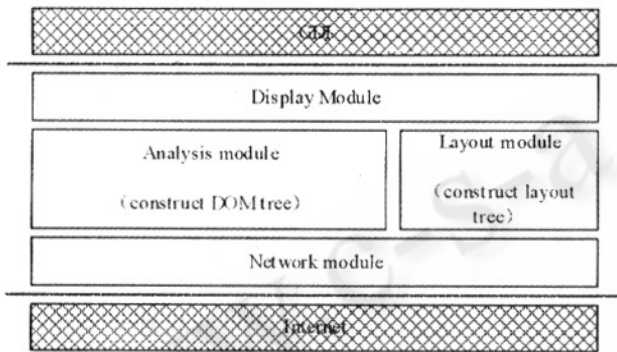


图 1

技术所带来的一系列的不兼容问题。W3C 制定了一系列的标准包括使用语言的规范,开发中应遵循的准则以及解释引擎的行为等,同时,W3C 还制定了包括 CSS、XML 等在内的一系列影响深远的标准。不过,由 W3C 所制定的 WEB 规范仅是推荐性质而非强制性的。许多网站并未完全遵循由 W3C 制定的一整套规范。

#### 3.2 DOM 规范

DOM(Document Object Model),文档对象模型。它描述了处理网页内容的接口和方法,把整个页面规划成节点层级的文档,是 HTML 和 XML 的 API。DOM 规范是与语言无关的。就目前而言,DOM 规范分为三个层级:Level1,Level2,Level3。其中,Level1 包括 DOM Core 和 DOM Html,DOM Core 提供了基于 XML 的文档结构图,DOM Html 则添加了一些 HTML 专用的对象和方法,从而扩展了 DOM Core。Level2 则引入了视图、事件、样式、遍历和范围等几个新模块。Level3 则引入了以统一的方式载入和保存文档的方法,DOM Core 也被扩展为支持所有的 XML 1.0 标准。

由于 DOM 并非一个强制性的标准,所以各浏览器对其的支持各不相同。当前的主流浏览器中,Mozilla

支持几乎所有的 Level 2 以及部分的 Level3,Opera 与 Safari 支持几乎所有的 Level1 与大部分 Level 2,IE6 支持大部分的 Level1。

### 4 浏览器框架结构

此浏览器分为几个大的模块,其分别是网络连接模块、语法规义分析模块、页面元素排版模块、图形图像显示模块等几大部分。其结构图如图 1 所示。

该浏览器的工作流程大致如下:首先,网络连接模块通过输入的网址从网络上获取数据;然后,将获取的数据传送给语法与语义分析模块。语法与语义分析模块主要做的工作有两个,其一是将整个 HTML 的文档解析成为一系列的分词,其二则是将这些分词构建成为一棵文档对象树,分词在对象树中的位置以及彼此之间的关系依赖于其在原文档中的位置与作用;再次,页面元素排版模块根据文档对象树,来确立文档内各元素在页面显示中的位置,大小,形状等一系列与排版相关的状态,生成一棵页面元素排版树。页面元素排版树的生成依赖于文档对象树;最后,图形图像显示模块根据文档对象树以及页面元素排版树所提供的信息,将 HTML 文档绘制出来。图形图像显示模块依赖于所采用的操作系统提供的图形图像函数库。

#### 4.1 网络连接模块

网络连接模块的主要工作就是负责浏览器与网络的连接以及一系列与网络连接相关的工作,用户键入网站网址,网络连接模块首先对输入的 URL 地址进行分析,并做一些必要的纠错处理,然后向相应的网站发送请求,获取网页内容。在此过程中,该模块还必须要处理网页的重定向等,而获取的网页内容将交由语法规义分析模块进行处理。

为了达到良好的网络连接效果,此部分可以采用了以下一些手段:使用多线程下载网页中的图片等资源。由于网页的内容,本身是由 HTML 文本以及一系列的图片、表格等资源组成,采用多线程下载的方式,在获取 HTML 文本的同时,也可以同时获取网页上的图片与表格等资源,大大的提升浏览器的连接效率;使用一些必要的 URL 纠错手段。用户在输入 URL 时,常常会因为不小心而伴随着一些错误,例如将“www”误为“ww”。像这样一些错误,可以通过对输入的 URL 字符串进行分析,自动更正。

## 4.2 语法规义分析模块

语法规义分析模块主要完成两个工作,首先是对获得的 HTML 文档进行分析将文档内容解析为一个一个的分词,这些分词一般是由 HTML 文档元素,属性,以及相应的文本内容构成。在解析的过程中,一些含有属性的元素标签会被分解为若干个分词。以下面这段 HTML 代码为例:

```
<HTML >
<HEAD >
<TITLE > 早间新闻 </TITLE >
</HEAD >
<BODY >
<A href = " www. sina. com. cn " > 早间新闻 </
A >
</BODY >
</HTML >
```

语法分析会首先将 HTML、HEAD、TITLE、早间新闻、/TITLE、/HEAD、BODY 解析成为单个的分词,在遇到有属性的元素时,例如 <A href = " www. sina. com. cn " >, 会将 A 和 href 解析为两个不同的分词,如果该元素具有多个属性,则每个属性都会成为一个独立的分词。

而解析的方式也可以有所不同。就目前而言,解析的方式有两种,其一是边下载边解析,其二则是等待 HTML 文档完全下载以后再进行解析。这两种解析方式各有其优缺点。第一种解析方式的主要优点在于,占用内存较少。不需要一个很大的缓冲空间来存放整个 HTML 文档,在下载的过程中,下载一部分即解析一部分,对于内存资源紧张的系统来说,比较合适。然而,此种方法的缺点在于大大提高了解析的难度。采用此种方法解析,需要解决的几个问题,第一,是要防止截取的数据块不完整。这种情况一般出现在截取的字符串的尾部,譬如将一个元素标签从中间截断。对于这个问题,一般可用回溯法,找到一个能确保截取块的完整性的字符,例如字符 '>'。第二个需要解决的问题,则是需要记录更多的上下文信息。由于此种解析方式,并不会保留完全的文档,而且每次接收的字符串在文档中的位置也不固定,所以相关的信息都需要保留,以便在后面的数据处理中使用,这也在一定程度上加大了后期处理的难度。而第二种解析方式,即等待 HTML 文档下载完成后再进行解析的方式,则与第

一种方式相反。等待文档全部下载到本地缓冲区,需要更多的时间以及占用更多的内存资源。然而其优点在于,后期的处理则相对比较简单,直接对本地缓冲中的文档进行处理即可,由于保存有完整的 HTML 文档,所以也不存在第一种解析方法所需要解决的问题。

在语法解析进行的同时,语义解析也同样在进行。如前所述,语法解析主要是将文档形式的 HTML 网页内容解析为一个一个单独的分词,而语义解析,则是根据 HTML 文档的内容,将这些单独的分词以一定的形式组织起来,以便于以后的分析和处理。而这样的组织形式,就是先前文中所提到的 DOM(文档对象模型)。当然,也可以不采用 DOM 的形式,但是,由于 DOM 对于动态显示 HTML 提供了很好的支持,并且具有良好的灵活性与可扩展性,因此,使用 DOM 构建文档对象树的方法在条件允许的情况下,相对来说更佳。

以上文中一段 HTML 代码为例,其解析结果如图 2 所示。

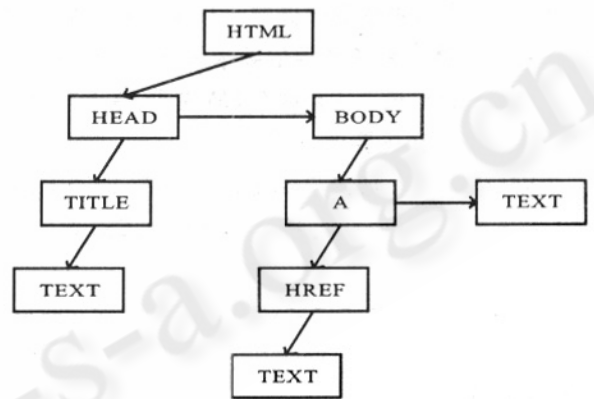


图 2

整个文档对象树根节点的子节点即是 HTML 节点,每个节点都有子节点与兄弟节点,子节点包含下层 HTML 元素以及当前 HTML 元素的属性,兄弟节点则是同一层次(即无包含关系)的 HTML 元素。

## 4.3 页面元素排版模块

页面元素排版模块,主要是根据已经生成的 HTML 文档对象树,分析各个 HTML 元素及其属性,生成一棵与界面相关的排版树。其各节点应当与文档对象树中的节点相对应,并应根据显示界面的情况,确定各节点显示的位置,大小,颜色等。排版树与文档对象树应当是相关的,当文档对象树发生变化时,排版树也应相应

的变化。这在显示动态 HTML 时是很重要的。

#### 4.4 图形图像显示模块

图形图像显示模块,其主要工作有两点,其一是决定当前的显示内容,其二则是运用由图像函数库所提供的图像绘制函数将当前应显示的内容绘制到显示屏幕上。

在本文中,由于是基于 WINCE 系统开发,所以其采用的函数库是由微软提供的 GDI 库函数。在每一次用户改变显示区域或者排版树发生改变时,图形图像显示模块都必须根据改变的内容,对显示屏幕上相应的区域进行重绘。

### 5 总结与进一步工作

在上述设计的基础上,目前已基本实现了一个基于 WINCE 系统的简单的嵌入式浏览器的构建。此浏览器基本具备了以下功能,下载 HTML 文件,并对其进行解析,生成相应的文档对象树,通过排版算法得到合适的显示结果,最后由图形图像显示模块绘制到显示屏幕上。该浏览器对系统资源的要求较低,其本身大小约为 200K,运行时最大占用内存 2M 左右,基本符合所应用的嵌入式系统的要求。

但就目前而言,此浏览器还并不完善,对于 DOM Level1 尚不能完全支持,而对于 javascript, CSS 等则完

全不支持。这是需要在下一步的工作中进一步完善的,也是后续工作的要点。另外就是对于语法语义算法以及排版算法的优化,以进一步减少该浏览器对系统资源的需求,并加快浏览器的解析过程。

#### 参考文献

- 1 Document Object Model (DOM) Level1 Specification. [http://www.w3.org/TR/1998/REC-DOM-Level-1-19981001/\[EB/OC\].](http://www.w3.org/TR/1998/REC-DOM-Level-1-19981001/[EB/OC].) W3C, 1998.
- 2 D. Ragget. HTML 4. 01 Specification W3C [M]. Dec1999.
- 3 张晓林、崔迎炜,嵌入式系统设计与实践,北京:航空航天大学出版社,2006.
- 4 陆云昆、于健、赵力、邹采容,嵌入式浏览器 Konqueror 的移植和汉化,电子设计应用,2006,12.
- 5 郭强、李洪涛、赵金钟、叶忠强、陈榕、周立柱,嵌入式浏览器架构研究与实现,计算机应用与软件,2007,24(2).
- 6 Yoshinori Saida, Hiroshi Chishima, Satoshi Hieda, Naoki Sato, Yukikazu Nakamoto. An Extensible Browser Architecture for Mobile Terminals[C]. Proceedings for the 24th International Conference on Distributed Computing Systems Workshops. 2004 IEEE.