

一个动态超媒体映射引擎*

钱培德 吕强 杨季文 朱巧明

(江苏省计算机信息处理技术重点实验室 苏州 215006)

(苏州大学计算机工程系 苏州 215006)

摘要 文章首先介绍了传统应用软件和超媒体应用软件的差异,指出了超媒体应用系统的特征和对最终用户的吸引力.在此基础上,建立了一个动态超媒体映射引擎的模型,它可以透明地为大多数传统应用系统增加超媒体的界面.文章最后给出了在 WWW 上实现的一个 DHyME(dynamic hypermedia mapping engine)实例.

关键词 超媒体,映射引擎,联系.

中图法分类号 TP311

超媒体(hypermedia)应用系统已经成为描述一个信息系统的新模式,超媒体和超文本(hypertext)在当前的学术界已经不加区分.超媒体的魅力不仅在于 media 这个词上:它把信息对象从传统的文字扩展为图像、声音等更接近于客观对象的形式,更重要的是,超媒体的魅力体现在 hyper 的含义上,即着重表达信息对象之间的自然联系,并以这种联系作为用户访问应用系统的基石.于是,用户通过超媒体应用系统,可以享受到更接近客观事实的、使用上更灵活的信息系统.

INTERNET 的普及和应用,为超媒体应用系统的推广和普及作出了贡献,但同时也为超媒体应用系统的更深层次、更高效率的推广制造了误解.通常人们倾向于认为在 WWW 上感受到的就是超媒体应用系统的全部.事实上,WWW 上的应用系统仅仅体现了超媒体应用系统的一小部分特征,远不是超媒体应用系统的全部.超媒体应用系统并不等于采用 HTTP 的 Web 应用程序.

本文首先从用户视图的观点指出了传统信息应用系统和超媒体应用系统的特征,从中比较了它们的差别.然后小结了超媒体的特点,指出超媒体系统的魅力所在.我们还建立了一个动态超媒体映射引擎 DHyME(dynamic hypermedia mapping engine)的模型,它可以透明地为大多数传统应用系统增加超媒体的界面,这些传统的应用系统包括采用 C/S 模型的应用系统.本文最后简单地给出了在 WWW 上实现的一个 DHyME 实例.

1 超媒体和超媒体应用系统的特征

1.1 传统应用系统的特征

传统的应用系统均可抽象为面向解决特定问题的信息检索系统.从用户视图来看,是将一个特定领域的信息集合,通过各种用户界面表达给用户.这种用户界面可以是层层递进菜单的路径指定式,也可以是用户输入参数的计算式.不管怎样,都是从特定的信息集合中抽取一部分,成为一个视图而提供给用户.

相对于超媒体而言,我们可以小结出以下关于传统应用系统的特征.

(1) 它面向一个特定的应用问题.

* 作者钱培德,1948年生,教授,博士生导师,主要研究领域为操作系统,中文信息处理.吕强,1965年生,副教授,主要研究领域为操作系统,中文信息处理技术.杨季文,1963年生,副教授,主要研究领域为中文信息处理技术.朱巧明,1963年生,副教授,主要研究领域为中文信息处理技术.

本文通讯联系人:钱培德,苏州 215006,苏州大学计算机工程系

本文 1998-06-05 收到原稿,1998-11-25 收到修改稿

(2) 用户界面是固定的,由应用系统的设计者在发布该应用系统之前就确定好了,即在系统的开发阶段就已确定,用户无权在使用的过程中调整。

(3) 由于用户界面的固定,导致该应用系统提供给用户的信息视图也是固定的,用户无法得到这些固定的视图之外的信息,哪怕是这些信息就囿于面向该特定问题的“信息资源”。

1.2 超媒体应用系统的特征

在超媒体应用系统中,首先,特定问题的“信息资源”用“元体(element)”来组织,虽然“信息资源”面向具体问题,但元体的组织和划分以表达这些客体本身的属性为主,建立元体与元体之间的联系(relationship)是超媒体系统的基石。

由于元体和联系的建立并不是只面向解决特定问题,所以联系可以有内联系(internal relationship)和外联系(external relationship)之分,它们分别表示应用系统内部的联系和跨越应用系统的联系。

用户通过用户界面得到的用户视图事实上是元体的展示,其特殊之处在于用户可以通过元体的联系来访问得到另外的视图,包括系统内的或系统外的,因此,有别于传统应用系统,超媒体系统的用户视图有以下特点。

(1) 由于外联系的存在,用户可以跨越系统得到相关信息,这里所谓的“相关”,既可以是客观事务本身的联系,也可以是主观建立的面向逻辑的联系,参见 1.3.3 节的标注特征。

(2) 由于用户视图的多样性,连开发者本身也无法罗列可能的用户视图,完全由用户在使用过程中的交互动态来决定。

(3) 传统应用系统提供用户视图树供用户遍历,树的路径均是确定的,而且只能在这棵树中遍历,超媒体系统提供一个无序的用户视图网络给用户遍历,遍历的路径几乎无数,遍历的节点可以不受限制,这是超媒体系统最大的优点,也是最大的缺点,好在有其他的理论和措施限制这个缺点^[1]。

1.3 超媒体的特征

超媒体的特征可以有多角度的描述,从应用系统的用户视图来看,其主要特征有结构(structure)、导航(navigation)和标注(annotation)3个方面^[2]。

1.3.1 结构特征

结构特征是超媒体的内在特征,也是联系的具体记载,在用户视图中,结构化的特征也是能够体会到的,如用户视图可以是一组具备语义的节点(semantically-typed node)集合,每个节点具备语义化的链接(semantically-typed link,名词),每个链接具备齐全的属性(link attribute),又如,用户视图可以是一个系统概述(overview)示意。

(1) 节点就是一个用户视图,它把若干个元体组合在一起,完成一定的逻辑功能,表现给用户,节点具备了语义功能后,就有了上下文的语境,从而可以体现出不同的语用来,简单地理解,具备了语义的节点,从内容上对节点进行了归类,帮助用户更快、更准确地找到用户视图。

(2) 链接的语义是在元体层次上进行内容上的分类,从节点过渡到节点是面向解决问题的,从元体连接到另外的部件则是事务本身联系的,所以,链接的语义同节点的语义属于不同的层次,超媒体系统可以有另外的机制为具体的用户限制或表现一定的链接语义,并非每个用户在任何条件下都可以享用所有链接的语义。

(3) 系统概述有全局与局部之分,从用户视图的角度来看,它们没有区别,只不过是用户视角的广和窄之分,系统概述实际上是节点语义的图形表示。

1.3.2 导航特征

导航特征是最容易体会的超媒体特征,也是超媒体系统区别于传统应用系统的最显著的特征,它是联系的外在体现,WWW 给用户最深刻的印象恐怕就是一部分的导航特征,如浏览、连结和部分索引。

导航特征可以细化为以下几点。

(1) 浏览,用 Microsoft 的术语叫 explore,用 Netscape 的术语叫 browse,均指显示或表现某个节点或元体的内容(可以是各种 media),这一点与传统的系统应用系统没有任何差别。

(2) 连结(动词),根据浏览到的节点或元体所提供的链接,访问其他与此联系的节点或元体,一般来说,具备连结功能的节点或元体在超媒体中被称为锚点(anchor)。

(3) 索引,这是将用户浏览的节点或元体索引起来,于是用户可以回到自己访问踪迹的任何一点。

(4) 查寻,这是连结的扩展,用户可以基于节点或元体,条件化地发出基于内容或基于结构的查寻,由应用系统过滤出可能的链接,再从这些链接中实施连结操作。

(5) 导图,这是索引的扩展,系统开发者或用户都可以组织一定的节点或元体,在其中建立一种固定的联系,用来完成一个具体的逻辑功能,其他用户就可以被限制在这个导图上,按指定的次序访问固定的内容。

1.3.3 标注特征

标注特征是指用户能够为浏览到的任何客体标上注释,或者强行为其建立本系统内、甚至本系统外的链接。传统应用系统的联系侧重的是应用系统内本身固有的联系,它针对解决一个固定的问题,是客体自身的记载,而标注特征则完全体现了用户的一种主观理解,它可以完全独立于客体本身的联系之外。

如果说超媒体的联系是开发者针对解决问题而建立,那么,标注特征提供用户一种能力,使他能够针对使用建立另外一种“联系”,或针对解决问题而增加在开发阶段未建立的“联系”。

标注特征和导航特征的基础都是结构特征,建立新的联系只能基于该客体的结构表达的基础,如果这种结构表达不能支持用户想建立的新联系,那么这种联系就无法建立。

2 动态超媒体映射引擎 DHyME 的设计

应该说超媒体应用系统并未在当前的应用系统中占据主流,因此,其魅力因技术和普及的原因并未得到充分展示。下面我们将建立一个动态超媒体映射引擎 DHyME 的模型,它可以透明地为非超媒体应用系统增加超媒体的界面,用户可以平滑地从传统应用系统过渡到超媒体系统,享受超媒体的功能。

2.1 目标系统的约束要求

相对来说,应用系统可以分为面向计算和面向文档两类,前者如会计结算系统、计算机辅助设计系统、专家系统、地理信息系统和统计分析软件包等;后者如资料检索系统、信息文档阅读系统。虽然这两种类型的应用系统在用户视图中没有任何差别,但是从实现的角度来说,面向文档的应用系统的主要工作在于文档的组织和用户界面;面向计算的应用系统除了实现用户界面来显示计算结果之外,还有一大部分工作在于针对问题建立模型及其实时计算。

对于面向文档的应用系统来说,增强超媒体功能的工作在于重新建立文档之间的联系,并在用户界面上实现对这些联系的访问,这是一种静态的工作过程。对于面向计算的应用系统来说,由于其在用户界面上显示的“文档”是一个动态的计算结果,所以为它建立超媒体功能的思路和方法均比较特殊。我们把这种以面向计算应用系统为特征的应用系统称为 DMIS(dynamically-mapped information system),意为不得不动态映射超媒体功能的信息系统^[1]。

DMIS 的特征是,应用系统明显地可以划分为显示模块和计算模块这样一前一后的两个组成部分。前者可以用 UIS(user interface system)来表示,后者还是借用 DMIS 来代表。不难理解,C/S 应用程序就是典型的 DMIS,Web 应用程序更是标准的 DMIS,甚至于静态的 Web 主页由于其 HTTP 框架的实现,也可以归类到 DMIS 中。

DMIS 就是 DHyME 工作的目标系统,DHyME 将为 DMIS 类型的应用系统透明地增加超媒体功能提供一种比较通用的解决方案。为 DMIS 增强超媒体功能的重要性和迫切性还在于:不同于面向文档的应用系统,DMIS 在界面上显示的“文档”具有更深的内涵,它的中间结果,例如这份“文档”的生成方法、生成参数等都具有重要价值,用超媒体的概念来实现和表示这类信息是最直接、最自然的方法。

2.2 DHyME的体系结构

DHyME 的总体目标是要在尽量少改动 DMIS 或根本不改动 DMIS 的基础上,为 DMIS 增加超媒体功能。图 1 是 DHyME 的结构示意图。

在图 1 中,DMIS 和 UIS 的概念在第 2.1 节中已经说明,下面我们来说明其他组成部分。

User Interface System Wrapper, 简称 UIW,是一个将通用的 DHyME 应用于一个特定应用程序界面方向的连接件。它的功能有 3 个:① 翻译转换 DHyME 和 UIS 之间的信息格式;② 实现 DHyME 和 UIS 之间的信

息流通;③ 实现 UIS 未能实现的、DHyME 所要求的超媒体用户界面。

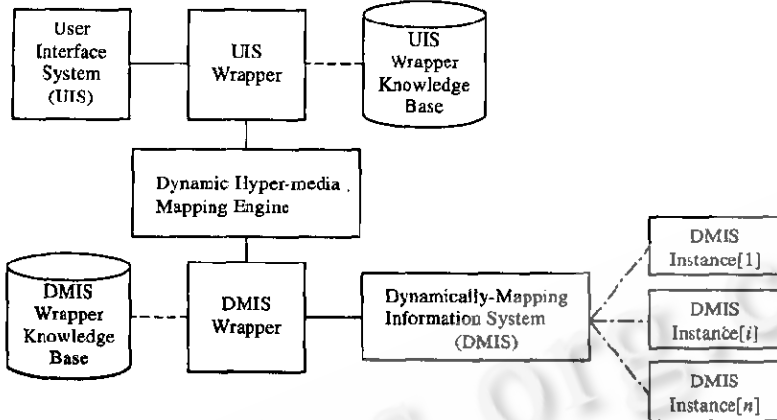


图 1 DHyME 的结构示意图

UIS Wrapper Knowledge Base, 含有所有 UIW 所需的与 UIS 通信用信息, 如格式转换信息、通信协议及例程、DMIS 和 UIS 的协调关系。

DMIS Wrapper, 简称 DMISW, 是一个将通用的 DHyME 应用于一个特定应用程序计算核心方向的连接件。它的功能除了实现 DHyME 和 DMIS 之间的信息转换和流通之外, 更重要的是, DMISW 的功能是要为来自 DMIS 的文档标注出可以具备超媒体功能的元体, 并将这些元体可以具备的联系一一准备好, 包括该应用程序 DMIS 相关于解决本特定问题的内联系和帮助理解决特定问题的外联系。DMISW 是实现 DHyME 对 DMIS 的透明程度的关键, 必须充分利用 DMIS 所提供的 API 和 DMIS 开发人员的知识。

DMIS Instance 是一些基于同一个 DMIS 的应用系统实例, 例如, 基于同一个数据库服务器的实例数据库和基于同一个专家系统开发 SHELL 的具体专家系统。

DMIS Wrapper Knowledge Base, 记录和实现 DMIS 内部结构到超媒体结构的映射。我们将在下一节详细叙述这个部分。

Dynamic Hypermedia Mapping Engine, 这是各种超媒体引擎集合的总称, 它是超媒体特征的实现者, 事实上是超媒体的功能模块的组合。例如, 可以有连结引擎、索引引擎、导图引擎和查询引擎等等。UIS 和 DMIS 之间的信息流动均经过 DHyME 过滤处理之后, 不管是在界面上还是在用户交互的内容上, 都有了原 DMIS 所不含有处理超媒体特征。

从图 1 中我们可以看到, DHyME 是与应用程序无关的, 它可以服务于所有的 UIW 和 DMISW。DHyME 工作在超媒体的映射模型之中, 实现超媒体的各种功能。两个 wrapper 除了起接口的作用外, 更重要的是实例化映射模型中的各种表格和数据。因此, wrapper 是与应用系统相关的, 将目标应用系统的各种要素: 静态的属性和动态的行为映射为超媒体的元体、节点及其联系, 必要的话, 补充提供新的逻辑的或物理的元体、节点及其联系。

2.3 映射机制

映射机制要解决的主要问题是描述和实现超媒体的结构特征。总体上来说, 我们通过一种称为桥接规则 (bridge law) 的机制来实现。具体来讲, 有 3 种类型的桥接规则。

(1) 元体标定桥接规则 (identification bridge law)

这类规则主要用于从 DMIS 到超媒体的映射。它描述了如何把 DMIS 的基本信息单元映射并组织成超媒体的元体或节点。元体和节点具有许多基于超媒体的属性, 元体标定桥接规则一方面要填充这些属性, 另一方面还要描述锚点在 UIS 的属性: 显示位置、显示模式、锚点位置等。事实上, 它桥接的是超媒体对象和 DMIS 对象。

例如, 在一张电子数据报表 (spreadsheet) 中, 作为 DMIS 对象的每个 cell 都可以是超媒体的元体, 它所具有的链接可以提供该 cell 是如何生成的实例; cell 所对应的列标题和行标题可以提供本列或本行的公式

(formula)连结.

(2) 链接桥接规则(link bridge law)

这类规则主要用来描述 UIS 中的锚点所具备的链接如何映射到一个新的“文档”,这实际上是连结的静态数据资源.DHyME 中的“动态”含义的另一种诠释也就在于每个链接都是动态生成的.

例如,如果每个链接都可以用一组命令序列来实现,链接桥接规则就可以描述这组命令序列.在这些命令中,可以有如下种类:

- ① DMIS 原有的内部命令,只不过 DMIS 把这些命令用在实现其他用户视图上;
- ② DMIS 的 API,DMIS 原来是准备给 UIS 或二次开发 DMIS 用的;
- ③ 由 DHyME 实现的和②的组合;
- ④ 其他 DMIS 的①和②;
- ⑤ 由 DHyME 实现的不同 DMIS 的内部命令和 API 的组合.

(3) 再现桥接规则(regeneration bridge law)

由于结构化的特征允许用户随机地访问元体或节点,因此,元体或节点的再现需要用特定的规则来描述,这就是再现桥接规则所要解决的问题.因为用户视图或节点有一定的上下文环境,当游离于这种上下文关系随机访问某个视图或节点时,不能要求用户继续给出这种上下文环境.再现桥接规则就是用来记录和实现每一个节点或元体的上下文环境,以便独立地产生相应的元体或节点.

例如,用户输入一组参数,得出一个方程解,又将这一组方程解组合另外的方程解表现在一张图示中.那么,当用户在别的视图中突然想要访问这个方程解,或者这张综合图示时,再现桥接规则能够使用户直接方程解的视图或综合图示.

3 一个实例

3.1 概述

我们在 DHyME 的指导下,在 WWW 上实现了一个超媒体引擎^[4],它可以动态地为浏览器用户提供新的多重链接:为同一个锚点提供不止一个链接,而这种新增加的超媒体功能不需要修改任何原有的 DMIS 和 UIS.其大致思路是:设置一个代理服务器(proxy server),当浏览器以此代理在 WWW 上发出请求时,代理服务器首先截获返回的文档,将此文档转发给 DMISW.DMISW 利用元体标定桥接规则把该文档内的所有元体标注出来,再将此文档递交给 DHyME.DHyME 根据此次请求用户的级别判定该为这些元体提供哪些链接,然后再将这份文档传递给 UIW.UIW 需要把具备链接的元体嵌入到文档中,使其成为锚点,并将控制机制嵌入文档,以便当用户操作这些锚点时,由 UIW 接受控制而不是由浏览器接受控制.最后,UIW 将这一份新文档返回给代理服务器,由其返回给浏览器作为用户请求的响应.

当用户操作到 DHyME 所支持的锚点时,UIW 先于浏览器截获控制,向 DHyME 发出链接请求.DHyME 根据链接桥接规则和当前用户的级别给出一组链接,UIW 就把这一组链接提供给用户选择.当用户选择了其中一个实施连结时,UIW 就会把连结命令传递给 DHyME.DHyME 取出链接桥接规则中的命令序列,依次发给相应的 DMIS 动态执行这些命令.当这些命令产生新文档时,又被相应的 DMISW 截获.如此循环往复.

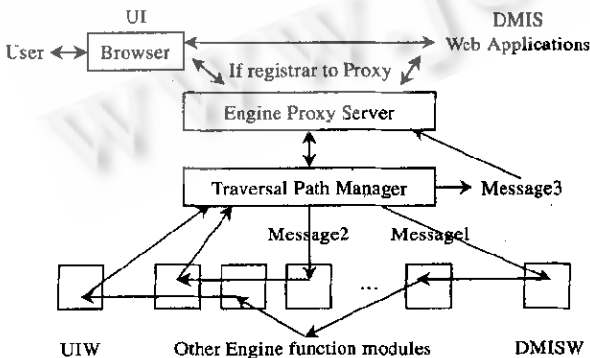


图2 DHyME 的功能模块和数据流示意图

图2是 DHyME 的功能模块和数据流示意图.

UI-DMIS 是一个完整的应用程序.代理服务器(engine proxy server)的作用是截获来往于 UI 和 DMIS 这

两个模块之间的消息,把这些消息放到引擎的各个功能模块中去处理.对于我们实现的原型来说,代理服务器就是 HTTP 代理服务器.

在 DHyME 的结构中,引擎功能模块(engine function module)是实现具体功能的载体,事实上,除了 DMIS 和 UIW 之外,其他模块均为引擎功能模块.这些功能模块可以分布地运行在 INTERNET 的各个主机上.其功能模块之间相互独立,彼此不可见.消息旅行定位管理器(traversal path manager)负责动态地确定所有消息流经功能模块的路径.如 Message3 由它判定了不需经过任何其他功能模块,直接流通于 UI 和 DMIS 之间;而 Message1 和 Message2 被判定为分别流过若干个功能模块,然后再回到 UI-DMIS 的循环.

3.2 消息对象设计

在 DHyME 中,消息采用了 XML(extensible markup language)规范,并采用 Microsoft 的 msxml 软件包实现了文档对象模型 DOM(document object model).于是,每个功能模块输入和输出的均为消息对象(在 DHyME 中被命名为 EngineMessage).对消息的产生和消费均按照面向对象的访问规范.

下面是一个消息对象的例子,这是一个由代理服务器产生的,来自于 Web 服务器的消息.

```

<!-- DHyME message By Qiang Lv -->
<?XML version= "1.0" ?>
<!DOCTYPE DOCUMENT [
<!ELEMENT DOCUMENT (Subject,MessageID,TraversalPath,MessBody)>
<!ELEMENT Subject (#PCDATA)>
<!ELEMENT MessageID (Owner,RealMessageID,MessageType,SubModules)>
<!ELEMENT Owner (#PCDATA)>
<!ELEMENT RealMessageID (#PCDATA)>
<!ELEMENT MessageType (#PCDATA)>
<!ELEMENT SubModules (#PCDATA)>
<!ELEMENT TraversalPath (#PCDATA)>
<!ELEMENT MessBody (#PCDATA)
]>
<DOCUMENT>
  <Subject>SourceDocument</Subject>
  <MessageID>
    <Owner>OriginalUser</Owner>
    <RealMessageID>doc#921213</RealMessageID>
    <MessageType>sample</MessageType>
    <SubModules>Proxy</SubModules>
  </MessageID>
  <TraversalPath>fun1+fun2*fun3+fun4</TraversalPath>
  <MessBody>
    <![CDATA[
      <HTML>
        ...
      </HTML>
    ]>
  </MessBody>
</DOCUMENT>

```

功能模块将此消息滤过 msxml 软件包后,就变成对消息对象的处理了,没有必要自己扫描消息来获取需要的信息.TraversalPath 元素的内容指示了本消息需要旅行功能模块 fun1,然后可以同时并行发给 fun2 和

fun3,最后再旅行到 fun4.

除了连结引擎以外,我们还实现了索引引擎,把用户访问的每个页面用再现桥接规则记载下来,用户可以随机访问任何一个被索引的页面.

4 结 语

DHyME 的贡献在于,建立一个平滑的、从传统应用系统到超媒体系统的过渡机制,使最终用户能够在传统应用系统的背景下享用超媒体的功能,为新的应用系统的开发提出合理的需求.同时,DHyME 的各个引擎的设计和实现,也正是超媒体功能的设计和实现,与具体的应用程序无关.所以,DHyME 的研究和进步同时也为超媒体系统的研究和开发作出了贡献.

参考文献

- 1 Conklin E J. Hypertext: a survey and introduction. *IEEE Computer*, 1987,20(9):17~41
- 2 Bieber M, Kacmar C. Designing hypertext support for computational applications. *Communications of the ACM*, 1995,38(8): 99~107
- 3 Bieber M, Vitali F. Toward support for hypermedia on the World Wide Web. *IEEE Computer*, 1997,30(1):62~70
- 4 Lü Q, Ramanathan P, Pattabiraman M *et al.* Automatically applying hypermedia to existing web applications. In: *Proceedings of Hypertext'98*. Pittsburg. 1998. <http://www.ics.uci.edu/pub/kanderso/ht98demos/lu.html>

A Dynamic Hypermedia Mapping Engine

QIAN Pei-de LÜ Qiang YANG Ji-wen ZHU Qiao-ming

(*Jiangsu Key Laboratory of Information Processing Technology Suzhou 215006*)

(*Department of Computer Engineering Suzhou University Suzhou 215006*)

Abstract The differences between traditional applications and hypermedia applications are compared in this paper. The main features of hypermedia system and the attraction for non-hypermedia application users are introduced. A model of dynamic hypermedia mapping engine is built, which can enhance non-hypermedia application with hypermedia features with minimal or no changes to it. A prototype of DhyME (dynamic hypermedia mapping engine) is also presented in this paper.

Key words Hypermedia, mapping engine, relationship.