(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 105210050 A
(43)申请公布日 2015.12.30

(21)申请号 201480026661.X
(22)申请日 2014.03.26
(30)优先权数据
61/805,394 2013.03.26 US
61/805,416 2013.03.26 US
14/179,302 2014.02.12 US
(85)PCT国际申请进入国家阶段日
2015.11.10
(86)PCT国际申请的申请数据
PCT/KR2014/002548 2014.03.26
(87)PCT国际申请的公布数据
WO2014/157938 EN 2014.10.02
(71)申请人 三星电子株式会社
地址 韩国京畿道
(72)发明人 林荣权 I. 布阿齐兹
(74)专利代理机构 北京市柳沈律师事务所 11105
代理人 邵亚丽 蔡军红
(51)Int.Cl.
G06F 17/00(2006.01)
G06F 17/21(2006.01)

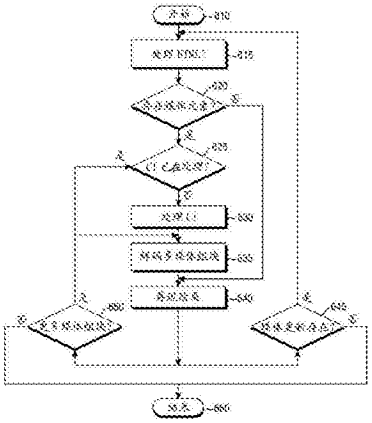
权利要求书1页 说明书12页 附图4页

(54)发明名称

用于呈现超文本标记语言页的装置和方法

(57)摘要

一种用于呈现HTML页的方法，包括：确定HTML文件是否包含对CI文档的参引，取回和处理描述至少一个HTML元素的行为的CI文档，并且基于该CI文档通过解码HTML文件呈现HTML页。一种用于呈现HTML页的装置，包括：处理电路，被配置成确定HTML文件是否包含对CI文档的参引，取回和处理描述至少一个HTML元素的行为的CI文档，基于该CI文档通过解码HTML文件呈现HTML页。



CN 105210050 A

1.一种用于呈现超文本标记语言(HTML)页的方法,包括:

确定HTML文件是否包含对CI文档的参引;

取回和处理描述至少一个HTML元素的行为的CI文档;

基于所述CI文档通过解码所述HTML文件呈现HTML页。

2.根据权利要求1所述的方法,还包括:

当检测到所述CI文档的更新时,基于更新后的CI文档重新呈现所述HTML页。

3.根据权利要求2所述的方法,其中,所述CI文档包括用于检测CI文档的更新的版本。

4.根据权利要求1所述的方法,其中,所述CI文档包括参引要播放的媒体组块的组块参考、和控制所述媒体组块的播放时间的同步单元(SU)。

5.根据权利要求4所述的方法,其中,所述CI文档包括多个SU,每个SU包括各自的参引每个媒体组块的组块参考。

6.根据权利要求4所述的方法,其中,所述SU被配置成提供用于播放每个媒体组块的开始时间。

7.根据权利要求4所述的方法,其中,所述SU被配置成提供用于播放多个媒体组块的每一个的相对于先前的SU的各自的相对时间。

8.根据权利要求4所述的方法,其中,所述CI文档被配置成提供关于所述至少一个HTML元素的空间布局的信息。

9.根据权利要求8所述的方法,其中,所述CI文档被配置成描述对至少一个HTML元素的风格的改变,所述风格包括如下至少一个:所述至少一个HTML元素的位置、外观、可见性。

10.根据权利要求1所述的方法,其中,所述CI文档包括参引要播放的媒体组块的组块参考和控制所述媒体组块的播放时间的同步单元。

11.一种用于呈现超文本标记语言(HTML)页的装置,该装置配置成在权利要求1到10中描述的方法中的至少一个。

用于呈现超文本标记语言页的装置和方法

技术领域

[0001] 本公开一般涉及用于呈现复合超文本标记语言 (HTML) 页的装置和方法，并且更具体地，描述在 HTML5 网页 (web) 文档中的元素 (element) 和多媒体组件的时间行为。

背景技术

[0002] 反映因特网上的多媒体服务的快速增加，下一代网页标准，HTML5 已包括通过引入两个新标签 (分别用于视频和音频的 <video> 和 <audio>) 来在网页上嵌入多媒体组件的标准化方式。两个元素提供了包括它们的一个或多个源多媒体数据的方式。对于 <video> 元素，也可以描述它的空间属性，诸如宽度和高度。

[0003] 这样的用于音频和视频的新标签没有提供描述性地表示对多媒体组件的时间行为的精确控制的方式。HTML5 假设使用 JavaScript 来编程控制多媒体组件的时间行为。HTML5 媒体 API 允许对用户呈现用户界面元素，以控制开始和暂停媒体元素。它也允许控制重放的速度和跳跃到媒体数据的特定位置。

发明内容

[0004] 技术问题

[0005] 但是，使用 JavaScript 来控制多媒体的时间行为有如下几个潜在缺点 ：由于 JavaScript 引擎不保证嵌入到 HTML5 页的脚本的实时处理，所以多媒体组件的时间要求严格的控制不能得到保证 ；由于多媒体组件的时间要求严格的时间行为和网页的其它组件的静态属性被混合在一个 DOM 树中，所以它们无法被单独处理，从而任何对 DOM 树的更新可能会延迟对多媒体组件的时间行为的时间要求严格的更新 ；以及由于脚本的生命周期通过嵌入其的网页加载来绑定，所以 HTML5 文件的任何更新或刷新将导致多媒体组件的再现的复位。

[0006] 技术方案

[0007] 一种用于呈现 HTML 页的方法，包括 ：确定 HTML 文件是否包含对 CI 文档的参引，取回和处理描述至少一个 HTML 元素的行为的 CI 文档，并且基于该 CI 文档通过解码 HTML 文件呈现 HTML 页。

[0008] 所述方法包括 : 当检测到 CI 文档的更新时，基于更新的 CI 文档重新呈现 HTML 页。

[0009] CI 文档包括用于检测 CI 文档的更新的版本。

[0010] CI 文档包括参引要播放的媒体组块的组块参考，以及控制媒体组块的播放时间的同步单元 (SU)。

[0011] CI 文档包括多个 SU，每个 SU 包括各自的参引每个媒体组块的组块参考。

[0012] SU 被配置为提供用于播放每个媒体组块的开始时间。

[0013] SU 被配置为提供用于播放多个媒体组块的每一个的相对于先前的 SU 的各自的相对时间。

[0014] CI 文档被配置为提供关于至少一个 HTML 元素的空间布局的信息。

[0015] CI 文档被配置为描述对至少一个 HTML 元素的风格的改变，该风格包括如下至少一个：至少一个 HTML 元素的位置、外观、可见性。

[0016] CI 文档包括参引要播放的媒体组块的组块参考、以及控制媒体组块的播放时间的同步单元。

[0017] 一种用于呈现 HTML 页的装置，包括：处理电路，被配置来确定 HTML 文件是否包含对 CI 文档的参引，取回和处理描述至少一个 HTML 元素的行为的 CI 文档，并且基于该 CI 文档通过解码 HTML 文件呈现 HTML 页。

[0018] 一种用于呈现 HTML 页的装置，包括：HTML 处理电路，被配置成确定 HTML 文件是否包含对 CI 文档的参引，媒体处理单元，被配置成取回和处理描述至少一个 HTML 元素的行为的 CI 文档，以及呈现单元，被配置成基于该 CI 文档通过解码 HTML 文件来呈现 HTML 页。

[0019] 在开始以下具体说明前，阐述贯穿该专利文件所使用的特定词语和词组的定义将是有益的，术语“包括”和“包含”以及它们的衍生词，表示没有约束的包括；术语“或者”包含和 / 或的含义；词组“相关联”和“对其关联”以及它们的衍生词可以意味着包括、包括在内、相互连接、包含、被包含在内、连接到或与…连接、耦合到或与…耦合、与…通信、与…协作、交织、并列、近似于、绑定到或与…绑定、具有、有…属性等；而术语“控制器”意味着控制至少一种操作的任何设备、系统或其部分，此类设备可以实现为硬件、固件或软件或者它们中至少两种的组合。应该注意与任何特定控制器相关联的功能可以是集中式或分布式的，本地的或远程的。贯穿该专利文件提供了某些词语和词组的定义，本领域普通技术人员应该理解，如果不是在绝大多数情况下，也是在许多情况下，此类定义适用于如此定义的词语和词组的现在以及将来的用法。

附图说明

[0020] 为了更完整地理解本公开及其优点，现在参考下面结合附图的详细描述，其中相同参考数字表示相同部分：

[0021] 图 1 示出根据本公开一实施例的无线网络；

[0022] 图 2 示出根据本公开实施例的示例组成信息 (CI) 层；

[0023] 图 3 示出根据本公开实施例的 HTML5 文件和 CI 文件的结构；

[0024] 图 4 示出根据本公开另一实施例的 HTML5 文件和 CI 文件的结构；

[0025] 图 5 是示意说明根据本公开实施例的示例媒体呈现系统的高层框图；

[0026] 图 6 是示出根据本公开实施例的处理内容的示例操作的流程图；

[0027] 图 7 示出其中可以实现本公开各个实施例的示例客户端设备；

[0028] 图 8 示出包是逻辑实体以及其根据 MMT 内容模型的逻辑结构；和

[0029] 图 9 说明来自通过呈现信息 (PI) 文档提供的不同资产 (Asset) 的多媒体处理单元 (MPU) 的呈现的定时的示例。

具体实施方式

[0030] 下面讨论的该专利文献中的图 1 到 9 以及用来描述本公开原理的各种实施例仅通过举例说明的方式，而不应以任何方式理解为限制本公开的范围。本领域技术人员将理解的是，本公开的原理可以在任何适当布置的无线通信系统中实现。

[0031] 图 1 示出其中可实施本公开的各种实施例的一点对多点传输系统 100 的示例。在所示实施例中，系统 100 包括发送实体 101、网络 105、接收实体 110-116、无线传输点（例如，演进节点 B(eNB)、节点 B)，诸如基站 (BS)102、基站 (BS)103 以及其它类似的基站或中继站（未示出）。发送实体 101 经由网络 105 与基站 102 和基站 103 通信，其中该网络可以是例如互联网、媒体广播网络或基于 IP 的通信系统。接收实体 110-116 经由网络 105 和 / 或基站 102 和 103 与发送实体 101 通信。

[0032] 基站 102 对基站 102 的覆盖区域 120 中第一多个接收实体（例如用户设备、移动电话机、移动站、用户站）提供对网络 105 的无线接入。第一多个接收实体包括用户设备 111（其可以位于小商店 (SB)）、用户设备 112（其可以位于企业 (E)）、用户设备 113（其可以位于 WiFi 热点 (HS)）、用户设备 114（其可以位于第一居所 (R)）、用户设备 115（其可以位于第二居所 (R)）、用户设备 116（其可以是移动设备 (M)，诸如蜂窝电话机、具有无线通信功能的膝上型计算机、具有无线通信功能的 PDA、平板计算机等）。

[0033] 基站 103 对基站 103 的覆盖区域 125 中的第二多个用户设备提供对网络 105 的无线接入。第二多个用户设备包括用户设备 115 和用户设备 116。在示范实施例中，基站 101-103 可以使用 OFDM 或 OFDMA 技术（包括如本公开的实施例描述的用于呈现 HTML 页的技术）彼此通信以及与用户设备 111-116 通信。

[0034] 虽然在图 1 中仅描述了六个用户设备，但是可以理解系统 100 可以对另外的用户设备提供无线宽带和网络接入。应注意用户设备 115 和用户设备 116 位于覆盖区域 120 和覆盖区域 125 两者的边缘。用户设备 115 和用户设备 116 均与基站 102 和基站 103 两者通信并且可以被称为操作在切换模式，如本领域的技术人员公知的。

[0035] 用户设备 111-116 可以经由网络 105 接入语音、数据、视频、视频会议和 / 或其它宽带服务。在示范实施例中，用户设备 111-116 的一个或多个可以与 WiFi WLAN 的接入点 (AP) 关联。用户设备 116 可以是包括具有无线功能的膝上型计算机、个人数据助理、笔记本计算机、手执设备或其它具有无线功能的设备的多个移动设备的任何一个。用户设备 114 和 115 例如可以是具有无线功能的个人计算机 (PC)、膝上型计算机、网关或其它设备。

[0036] 图 2 示出根据本公开实施例的示例组成信息 (CI) 层 200。图 4 所示的实施例仅用于说明。能够使用其它 CI 层的实施例而不会脱离本公开的范围。

[0037] CI 层 200 被设计来例如使用声明的方式提供多媒体组件在 HTML5 网页上的时间行为。在某些实施方案中，组成信息拆分为两个独立的文件 :HTML5 文件 215 和组成信息 (CI) 文件 210。兼容的 HTML 5 文件 215 提供初始的空间布局和占位媒体元素 (place holder media element)，而组成信息文件 210 包含用于控制媒体呈现的定时指令。

[0038] 在某些实施例中，通过 CI 文件 210 来提供 HTML5 文件的网页上的多媒体组件的时间行为，其中该网页可以通过使用其 URI 来指定并且在 HTML5 文件中的多媒体元素（诸如 <video（视频）> 和 <audio（音频）> 元素）通过它们的 ID 来指定。换句话说，CI 文件 210 描述了为部分多媒体数据的时间行为的组合的多媒体组件的时间行为而不是多媒体数据的整个长度的时间行为来构造为来自多个数据的数据的灵活组合的多媒体组件。

[0039] 图 3 示出根据本公开实施例的 HTML5 文件 360 和 CI 文件 310 的结构。图 3 的实施例仅用于说明。能够使用其它实施例而不会脱离本公开的范围。

[0040] 在某些实施例中，可以通过在 CI 文件定义同步单元 (SU) 来表示多媒体组件的时

间行为。CI 文件 310 包括 SU 315，其包含一个或多个分别参引一个或多个媒体组块 340-1 到 340-n（分别包括多媒体数据）的组块信息 320-1 至 320-n。

[0041] 在采用绝对时间的某些实施例中，在 CI 文件 310 中的 SU 315 可以通过利用来自 W3C SMIL（同步多媒体集成语言）的公知属性（诸如 clipbegin（剪辑开始））提供组块 340-1 到 340-n 的按照时间的开始位置来指定特定时间而非该组块的开始，作为重放的开始点。

[0042] 在采用绝对时间的某些实施例中，SU 310 可以通过使用绝对时间（例如，UTC）提供列在首位的组块的开始时间或结束时间。此外，SU 320 可以使用相对于其它 SU 的相对时间或在 HTML5 网页中定义的事件（CI 文件将参考的）。换言之，在 SU 310 中列出的第一组块 340-1 的开始时间与 SU 310 的开始时间相同。在 SU 中所列的除第一组块 340-1 之外的组块 340-2 至 340-n 的开始时间与前一组块的结束时间相同，其中该结束时间是通过开始时间和持续时间之和给出。每个组块可能具有关于相对于它们所属的多媒体数据的开始的开始时间的某些信息，但这样的信息在本公开实施例中不用于同步。

[0043] 图 4 示出根据本公开另一实施例的 HTML5 文件 460 和 CI 文件 410 的结构 400。图 4 所示的实施例仅用于说明。能够使用其它实施例而不会脱离本公开的范围。

[0044] 在实施例中，HTML5 文件 460 可具有一个以上的多媒体组件并且 CI 文件 410 可以具有多于一个的 SU 420-1 至 420-n。在这种情况下，参引彼此不同的多媒体元素的两个或多个 SU 可以在相同时间中彼此重叠。

[0045] 或者，在 HTML5 网页 410 上的单个多媒体组件可以由一个以上的 SU420-1 至 420-n 参引。在本实施例中，通过来自一个以上的多媒体数据的组块来呈现多媒体元素。参引相同的多媒体组件的 SU 可能在时间上不彼此重叠。

[0046] 在某些实施例中，单个 CI 文件 460 被传递到多个客户端并且一些 SU 可以具有可以基于每个客户端的情况（诸如类型、位置、用户简档等）而被不同地解释的组块信息。这样的 SU 的持续时间可以基于相关联的组块的持续时间改变，使得其随后的 SU 的开始时间可能对每个客户端是不同的。

[0047] 在某些实施例中，公知的数据格式可用于媒体数据的组块，例如通过 HTTP 的动态自适应流 (DASH) 的段或 MPEG 媒体传输 (MMT)、媒体处理单元 (MPU)。在使用 MMT MPU 作为组块的数据格式的情况下，MMT 资产 ID 和每个 MPU 的序列号可用作引用特定的 MPU 作为组块信息的方法。

[0048] 在某些实施例中，公知的清单文件格式（诸如 DASH MPD（媒体呈现描述））可被用作组块信息。在使用 DASH MPD 作为组块信息的情况下，在 DASH MPD 列出的第一时间段的开始时间通过 SU 的开始时间来给定，而 SU 可以从这样的 MPD 中通过参引特定的时间段来提供 MPD 在时间中的开始位置。

[0049] 在某些实施例中，CI 文件 310 提供关于在 HTML 网页上的元素的空间布局和外观的信息。另外，CI 文件 310 可以提供关于该呈现的元素的空间布局的修改指令。在该实施例中，可以通过 divLayout 元素涵盖空间布局。

[0050] 关于 CI 的语法和语义，在某些实施例中，该组成信息可以基于 XML 的格式以信号通知该呈现事件并且更新客户端。在一种方法中，可以利用声明性类型的信令来格式化 CI 信息，其中 SMIL- 类似的语法用来表示特定的媒体元素的重放时间。声明性类型的信令还

用于通过指定要被显示在辅助屏幕中的特定“div”元素来指示辅助屏幕布局。

[0051] 利用声明性CI语法，将CI文件格式化为XML文件，其中该XML文件在语法上类似于SMIL。这种方法尝试尽量多地保留目前由MMT的第二委员会草案(CD)提供的解决方案。它提取了HTML 5扩展并将它们放入单独的CI文件中。该CI文件包含视图和媒体元素的序列。视图元素包含divLocation项的列表，这指示div元素在主HTML5文档中的空间位置。divLocation项的列表可以同时指向辅助屏幕。在主HTML 5文档中媒体元素指代具有相同标识符的媒体元素并且提供关于相应媒体元素的播放的开始和结束的定时信息。

[0052] 或者，媒体呈现的事件被提供作为改变媒体呈现的动作。这些动作可以很容易地被转换成JavaScript。

[0053] 利用基于动作的CI语法，CI文件被格式化为XML文件以简化其创作和加工方面的处理。如上所述，CI对从HTML5产生的在特定时间点的DOM树施加修改。

[0054] 该CI支持相对时间和绝对时间两者。相对时间使用文件的加载时间作为基准。绝对时间指的是时钟时间，并假定客户端例如使用NTP协议或通过其它方式与UTC时间时间同步。

[0055] 通过对该元素分配新的CSS样式执行对呈现的元素的空间布局和外观的定时修改。通过调用相应的媒体重放控制功能来控制媒体元素的重放。

[0056] 在下表中提供CI文档的XML模式(schema)：

[0057] 【表1】

[0058]

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns="urn:mpeg:MMT:schema:CI:2013" elementFormDefault="qualified"
attributeFormDefault="unqualified">
    <xs:annotation>
        <xs:appinfo>MMT Composition Information</xs:appinfo>
        <xs:documentation xml:lang="en">This Schema defines the MMT
Composition Information for MMT.</xs:documentation>
    </xs:annotation>

    <xs:element name="CI" type="CItype" />

    <xs:complexType name="CItype">
        <xs:sequence>
            <xs:element name="Action" type="ActionType"/>
            <xs:any namespace="##other" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:attribute name="version" type="xs:unsignedInt" />
        <xs:attribute name="clock" type="ClockType" />
        <xs:anyAttribute namespace="##other" processContents="lax"/>
    </xs:complexType>

    <xs:simpleType name="ClockType">
        <xs:restriction base="xs:string">
            <xs:enumeration value="UTC"/>
            <xs:enumeration value="relative"/>
        </xs:restriction>
    </xs:simpleType>

    <xs:complexType name="ActionType">
        <xs:sequence>
            <xs:element name="ActionItem" type="ActionItemType"/>
            <xs:any namespace="##other" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:attribute name="id" type="xs:unsignedInt" />
        <xs:attribute name="time" type="xs:duration" />
        <xs:anyAttribute namespace="##other" processContents="lax"/>
    </xs:complexType>

    <xs:complexType name="ActionItem">
        <xs:sequence>
            <xs:any namespace="##other" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:attribute name="type" type="ActionTypeType" />
        <xs:attribute name="target" type="xs:string" />
        <xs:attribute name="action" type="xs:string" />
        <xs:anyAttribute namespace="##other" processContents="lax"/>
    </xs:complexType>

    <xs:simpleType name="ActionTypeType">
        <xs:restriction base="xs:string">
            <xs:enumeration value="style"/>
            <xs:enumeration value="screen"/>
            <xs:enumeration value="update"/>
            <xs:enumeration value="source"/>
            <xs:enumeration value="media"/>
        </xs:restriction>
    </xs:simpleType>

</xs:schema>
```

[0059] 该CI由在特定时间施加的一组动作组成。动作项可以与该媒体或它的源、元素的风格、呈现的更新（例如，替换或者删除元素）或屏幕有关。

[0060] 几个动作项可以被捆绑在一起以用于在相同时间执行。每个动作项指定在DOM中的它将施加该动作的目标元素。例如，风格动作将施加在动作串中提供的@style到通过@target属性标识的DOM元素。媒体动作项包含将对通过@target属性标识的媒体元素执行

的媒体功能。

[0061] 在下表中提供 CI 元素的语义：

[0062] 【表 2】

[0063]

元素或属性名称	使用	描述
CI	M	CI 的根元素
@version(版本)	M	当前 CI 的版本。该版本字段被用于标识复制的 CI 信息
@clock（时钟）	OD 缺省关系	时钟提供在该 CI 中提供的时钟信息的类型。如果该属性缺少，则应当假定是相关的。在该情况下，该时间信息采用已装载文档的时刻。在其存在并且具有值“绝对”时，在该 CI 中的时间指示是 UTC 时间
Action(动作)	0...N	动作元素包含关于要施加于该 DOM 的动作的信息
@id	M	当前动作的标识符。具有相同标识符的动作被认为是相同的，即便是跨越多个 CI 文件
@time（时间）	M	要执行当前动作的时间
ActionItem(动作项)	1...N	动作可以包括几个动作项
@type(类型)	M	当前动作项的类型。它既可以是媒体动作、源动作、风格动作、更新动作，也可以是屏幕动作
@target(目标)	M	当前动作项将施加于的目标 DOM 元素
@action(动作)	M	携带对所述动作的描述的串。该串的值将基于所述动作的类型来解释

[0064] 下表定义对每个动作类型允许的可能的动作。

[0065] 【表 3】

[0066]

动作类型	可能动作
媒体	允许的动作是“播放”、“暂停”、“装载”
源	动作属性必须保持
屏幕	在 MMT 的第二 SoCD 的表 2 中定义了可能的动作
更新	允许的动作是替代在DOM中的目标元素的元素定义，如果目标元素标识符仍不存在，则新元素将被创建并拼接到所述 DOM 中。
风格	允许的动作是包含一个或多个 CSS 3 指令的串。该风格指令将在指定时间被应用于目标元素

[0067] 同时，CI 将稳步地要求更新，尤其是在现场服务的情况下。此更新可以由以下需求产生：更新媒体源，空间布局，添加或移除媒体元素，标记在辅助屏幕上的再现的内容等。

[0068] 对于这一点，CI 处理引擎应当观察 CI 的更新。这可以例如通过验证 CI 文件的版本号来实现。当检测到新 CI 时，它所包含的指令被提取并调度以用于执行。在一个典型的实现中，CI 处理引擎可以被实现为 JavaScript。该 JavaScript 可以通过从本地文件读取或使用 AJAX 定期取出 CI 文件。可替代地，MMT 接收器可以在每次它接收到新版本的 CI 文档时调用用于 CI 处理引擎的事件。

[0069] 下表是根据基于动作的格式创作的 CI 文件的例子：

[0070] 【表 4】

[0071]

```
<?xml version="1.0" encoding="UTF-8"?>
<CI clock="absolute" version="5"
        xsi:schemaLocation="urn:mpeg:MMT:schema:CI:2013 Untitled1.xsd"
        xmlns="urn:mpeg:MMT:schema:CI:2013"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

    <Action time="P43Y4M25DT10H30M35S" id="0">
        <ActionItem action="complementary" type="screen" target="div2"/>
    </Action>

    <Action time="P43Y4M25DT10H30M35.154S" id="1">
        <ActionItem action="http://www.example.com/asset1/mpu3.mp4"
type="source" target="video1"/>
        <ActionItem action="play" type="media" target="video1"/>
    </Action>

    <Action time="P43Y4M25DT10H30M36S" id="2">
        <ActionItem action=" " type="update" target="div3"/>
    </Action>


    <Action time="P43Y4M25DT10H30M45S" id="3">
        <ActionItem action="top=50px; left=100px;" type="style" target="div1"/>
    </Action>
</CI>
```

[0072] 在这个例子中，若干动作被指示。第一动作通知具有 id“div2”的“div”元素被建议仅在辅助屏幕上显示。因此，它将从主屏幕隐藏。第二动作包含 2 个动作项。第一动作项设置视频元素的源。第二项将启动视频项目在指定的动作时间处的播放。第三动作是更新动作，其完全地从 DOM 中移除具有 id“div3”的元素。最后动作是风格动作。它将目标元素“div1”的位置设置为 (50,100) 像素。

[0073] 为了与媒体呈现描述 (MPD) 协调，作为在 MPEG MMT 和 DASH（通过 HTTP 的动态自适应流）特设组 (adhoc group) 之间的正在进行的讨论的一部分，在 DASH 和 MMT 之间的协调的努力正被广泛讨论。DASH 基于 MPD，其作为描述访问内容和相关的定时的可能方式的清单文件。但是，MPD 不提供媒体呈现的先进特征。例如，它不提供控制媒体呈现的布局的工具。记住 MPD 还被设计有单个内容（包括其所有的媒体组件）的传递和呈现需求。目前无法提供具有重叠时间线或相关的时间线的不同的内容段作为单个 MPD 的一部分。因此，不能仅使用 MPD 来实现广告插入的几个使用情形。

[0074] 另外，MMT 的 CI 层目的在于解决类似的使用情形。它通过继承 HTML5 的能力以用于同时支持多个媒体元素和添加 CI 文件（它定义媒体呈现的动态）来这样做。通过使得能够进行根据 CI 的信息的 MPD 的寻址，媒体呈现可以容易地使用 DASH 传递。例如，这可以通过将媒体元素的源设置为指向 MPD 文件的指针来实现。

[0075] 由于在段 / 呈现层的协调，指向 MPD 可以促进媒体消费，因为将被处理的 MPU 将通过 MPD（作为一组连续时段的一部分，每个时段指向一个或多个作为表示 (Representation) 的 MPU) 来提供。

[0076] 使用 CI 文件还允许提供多个以及在时间上重叠的内容。这是通过将在 HTML5 中的多个媒体元素映射到多个 MPD 文件来实现的。每个媒体元件将被分配播放开始时间，其确定何时 MPD 的第一时间段的第一媒体段将被播放出来。下表是 CI 的示例，其参引 2 个 MPD 并使用 SMIL- 类似的语法：

[0077] 【表 5】

[0078]

```
<html>
    <body>
        <video id="video1" src="content1.mpd" begin="0" end="100"/>
        <video id="video2" src="advert1.mpd" begin="20" end="25"/>
    </body>
</html>
```

[0079] 【表 6】

[0080]

```
<?xml version="1.0" encoding="UTF-8"?>
<CI clock="absolute" version="5"
        xsi:schemaLocation="urn:mpeg:MMT:schema:CI:2013 Untitled1.xsd"
        xmlns="urn:mpeg:MMT:schema:CI:2013"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

    <Action time="P0S" id="1">
        <ActionItem action="content1.mpd" type="source" target="video1"/>
        <ActionItem action="play" type="media" target="video1"/>
    </Action>

    <Action time="P20S" id="1">
        <ActionItem action="advert1.mpd" type="source" target="video2"/>
        <ActionItem action="play" type="media" target="video2"/>
    </Action>
</CI>
```

[0081] 注意，该呈现时间指令覆写由 MPD 给出的时间呈现时间暗示（例如 availabilityStartTime（可用开始时间）+suggestedPresentationDelay（建议的呈现延迟））。

[0082] 图 5 是示意说明根据本公开实施例的示例媒体呈现系统 500 的高层框图。图 5 所示的实施例仅用于说明。能够使用媒体呈现系统的其它实施例而不脱离本公开的范围。

[0083] 该系统包括在上层的呈现引擎 510 和在下层的 HTML 处理引擎 520 和媒体处理引擎 530。HTML5 引擎 520 处理 HTML5 网页而媒体处理引擎 530 处理 CI 文件和在其中列出的组块。呈现引擎 510 合并媒体处理引擎 530 的结果和 HTML 处理引擎 510 的结果，并一起再现其。

[0084] 更具体地，HTML5 引擎 520 将 HTML 文件 5 文件解析成文档对象模型 (DOM) 树并存储到存储器中。

[0085] 媒体处理引擎 530 取回 CI 与 HTML5 文件（和任何其它引用的文件），并处理该 CI 信息以相应地控制呈现。

[0086] HTML 处理引擎 520 和媒体处理引擎 530 可以在不同时间更新它们的结果。在一些例子中，在 HTML 处理引擎 520 正解析 HTML5 文件并且构造再现树的同时，媒体处理引擎 530 可以不断地更新解码的媒体数据。对于更新 CI，媒体处理引擎 530 根据在 CI 文件中可用的指令在特定时间处对 DOM 施加改变。该 DOM 节点 / 元素使用它们的标识符或可能使用特定模式（例如，通过 jQuery 选择器提供）来引用。

[0087] 图 6 是示出根据本公开实施例的处理内容的示例操作的流程图。虽然流程图描绘了一系列的顺序步骤，但是除非明确说明，不应该从该序列中得到以下的推断，有关执行的特定的顺序，顺序执行的步骤或其一部分，而不是同时地或以重叠的方式执行，或明确描绘的步骤执行而不发生插入的或中间步骤。在所描绘的示例中所描绘的操作由 UE 中的处理电路执行。

[0088] 处理操作开始于操作 610。在操作 615 中，客户端设备处理 HTML5 网页，并评估是否有媒体元素。如果没有媒体元素，那么它前进到执行 HTML5 的处理结果的操作。此处，客户端设备可以通过任何合适设备，其能够与服务器通信，包括移动设备和个人计算机。

[0089] 如果在操作 620 中存在任何媒体元素，那么在操作 625 中客户端设备首先评估是否相同 CI 文件已处于处理中。如果它已经处于处理中，则进入解码多媒体组块的操作 635。如果不是，那么在操作 630 中它处理 CI 文档。

[0090] 在操作 630 中处理 CI 文件之后，在 CI 文件中列出的多媒体组块在操作 535 中被处理，然后在操作 640 中再现结果。可以重复这样的操作，直到不存在需要处理的多媒体组块。如果在操作 645 中有 HTML5 的任何更新存在，则在解码和再现被处理的组块期间，客户端设备在任何时间将其与对多媒体组块的处理并行处理。

[0091] 图 7 示出其中可以实现本公开各个实施例的示例客户端设备 700。在本例中，客户端设备 700 包括控制器 704、存储器 706、持久存储器 708、通信单元 710、输入 / 输出 (I/O) 单元 712 和显示器 714。在这些说明性的例子中，客户端设备 700 是图 1 的发送实体 101 和 / 或接收实体 110-116 的一个实施的例子。

[0092] 控制器 704 是控制至少一个操作的任何设备、系统或其一部分。这样的设备可以以硬件、固件或软件或者其中至少两种的一些组合来实现。例如，控制器 704 可包括被配置成控制客户端设备 700 的操作的硬件处理单元和 / 或软件程序。例如，控制器 704 处理可以加载到存储器 706 的软件的指令。控制器 704 可包括多个处理器、多处理器核心或一些其它类型的处理器，这取决于具体实现。此外，控制器 704 可使用多个异构处理器系统（其

中主处理器与从处理器存在于单个芯片上）来实现。作为另一说明性示例，控制器 704 可以包括包含相同类型的多个处理器的对称多处理器系统。

[0093] 存储器 706 和持久存储器 708 是存储设备 716 的示例。存储设备是能够存储诸如例如不限制于数据、功能形式的程序代码和 / 或其它合适信息之类的信息的任何硬件，无论是基于临时的和 / 或基于持久的。在这些例子中的存储器 706 例如可以是随机存取存储器或任何其它合适的易失性或非易失性存储设备。例如，持久存储器 708 可以包含一个或多个部件或器件。持久存储器 708 可以是硬盘驱动器、闪速存储器、光盘或上述的一些组合。持久存储器 708 使用的媒体也可以是可移除的。例如，可移除硬盘驱动器可用于持久存储器 708。

[0094] 通信单元 710 提供与其它数据处理系统或设备的通信。在这些例子中，通信单元 710 可包括无线（蜂窝，WiFi 等）发送器、接收器和 / 或发送器、网络接口卡和 / 或任何其它适当的硬件，用于在物理或无线通信媒介上发送和 / 或接收通信。通信单元 710 可以通过使用物理和无线通信链路中的任一个或两者来提供通信。

[0095] 输入 / 输出单元 712 允许与能够连接到的其它设备或客户端设备 700 的一部分的数据的输入和输出。例如，输入 / 输出单元 712 可以包括接收触摸用户输入的触摸面板、接收音频输入的麦克风、提供音频输出的扬声器和 / 或提供触觉输出的电机。输入 / 输出单元 712 是用于向客户端设备 700 的用户提供和传递媒体数据（例如，音频数据）的用户接口的一个例子。在另一示例中，输入 / 输出单元 712 可以通过键盘、鼠标、外部扬声器、外部麦克风和 / 或一些其它合适的输入 / 输出设备提供用于用户输入的连接。此外，输入 / 输出单元 712 可发送输出到打印机。显示器 714 提供向用户显示信息的机制，并且是用于向客户端设备 700 的用户提供和传递媒体数据（例如，图像和 / 或视频数据）的用户接口的一个例子。

[0096] 操作系统、公开内容或其它程序的程序代码可以位于与控制器 704 通信的存储设备 716。在一些实施例中，程序代码处于在持久存储器 708 上的功能形式。这些指令可以被加载到存储器 706 中以用于由控制器 704 进行的处理。可以由控制器 704 使用计算机实现的指令（其可以位于存储器 706）来执行不同实施例的过程。例如，控制器 704 可以执行一个或多个上述模块和 / 或设备的过程。

[0097] 图 8 示出根据本公开实施例的 MMT 内容模型。应当指出的是，CI 文件适用于呈现和媒体文件的任何格式。在图 8 的实施例仅用于说明。可以使用 MMT 内容模型的其它实施例而不会脱离本公开的范围。

[0098] 包是逻辑实体，并且其在 MMT 内容模型中的逻辑结构示于图 8 中。在下文中，将定义作为编码的媒体数据和用于传递和消费目的相关联信息的集合的包的逻辑结构。

[0099] 首先，一个包应包含一个或多个呈现信息文档（诸如在 MMT 标准的部分 11 中规定的一个）、可能具有相关的传输特征的一个或多个资产 (Asset)。资产是共享相同资产 ID 的一个或多个媒体处理单元 (MPU) 的集合。资产包含编码的媒体数据，诸如音频或视频、或网页。媒体数据既可以是定时的，也可以是不定时的。

[0100] 呈现信息 (PI) 文档指定在用于消费的资产之间的空间和时间关系。在该标准的部分 11 中规定的 HTML5 和组成信息 (CI) 文档的组合是 PI 文档的例子。PI 文档也可以用来确定包中的资产的传递顺序。PI 文档要么将作为在该规范中定义的一个或多个消息传

递，要么通过本规范中未规定的一些手段作为完整文件被传递。在广播传递的情况下，服务提供商可以决定传送带呈现信息文档并且确定执行传送的频率。

[0101] 其次，资产是用于建立多媒体呈现的任何多媒体数据。资产是 MPU 的逻辑分组，其共享同一资产 ID 以用于携带编码的媒体数据。资产的编码的媒体数据既可以是定时数据，也可以是非定时数据。定时数据是具有固有的时间线并且会要求在指定时间的数据单元的同步解码和呈现的编码的媒体数据。非定时数据是可基于服务的上下文或来自用户的指示在任意时间被解码的其它类型的数据。

[0102] 单一资产的 MPU 应当具有定时或不定时的媒体。携带定时的媒体数据的相同的资产的两个 MPU 应当在其呈现时间没有重叠。在不存在呈现指示时，相同的资产的 MPU 可根据它们的序列号顺序地播放。

[0103] 任何类型的可以由 MMT 接收实体的呈现引擎单独消耗的媒体数据都被认为是单独资产。可以被认为是单独资产的媒体数据类型的示例为音频、视频或网页。

[0104] 在一些实施例中，以上描述的各种功能由计算机程序产品实现或支持，其中该计算机程序产品由计算机可读程序代码形成并且被包含在计算机可读介质上。用于计算机程序产品的程序代码可以以功能形式位于计算机可读存储设备上，其中该计算机可读存储设备是选择性地可移除的，并且可以被加载到或传递到客户端设备 700 以用于由控制器 704 进行的处理。在一些说明性的实施例，程序代码可以从另一设备或数据处理系统通过网络下载到持久存储器 708 以用于在客户端设备 700 中的使用。例如，存储在服务器数据处理系统的计算机可读存储介质中的程序代码可以通过网络从该服务器下载到客户端设备 700。提供程序代码的数据处理系统可以是服务器计算机、客户端计算机或能够存储和传送程序代码的某些其它设备。

[0105] 根据本公开的实施例提供基于 MMT 中的 CI 功能的解决方案。所述实施例将解决关于 HTML 5 扩展所关注的问题并会提供灵活的框架，其中所述框架对实现者提供使用合适技术的许多自由（例如，它可以基于 JavaScript 或本身来实现）。

[0106] 虽然已经利用示范实施例描述了本公开，但是，可以对本领域技术人员建议各种修改和改变。本公开意图涵盖这样的落在所附权利要求的范围内的修改和改变。

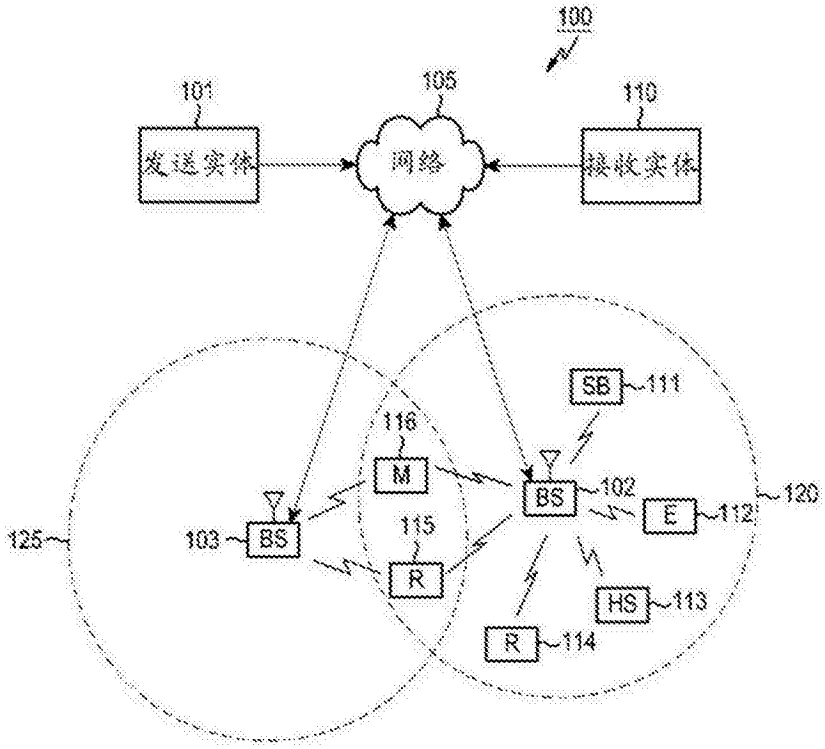

100
101
105
110
发送实体
网络
接收实体
SB
111
116
M
BS
102
E
112
120
125
103
BS
115
R
HS
113
R
114

图 1

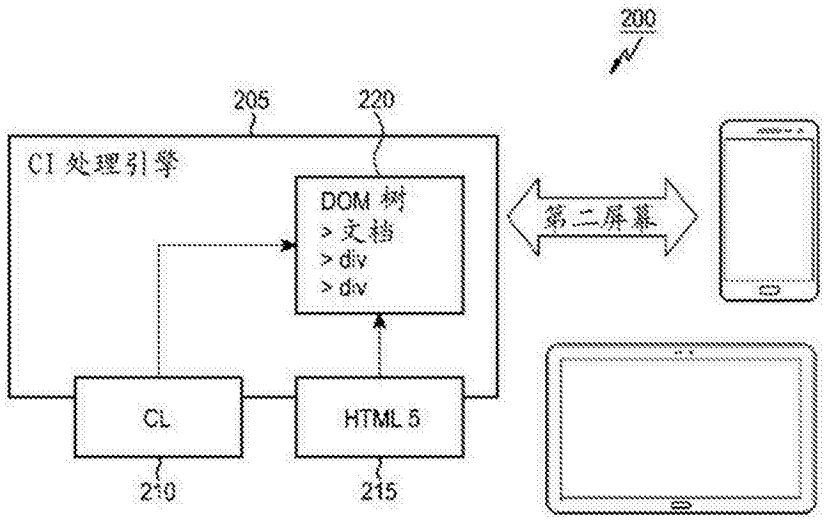

200
205
220
CI 处理引擎
DOM 树
> 文档
> div
> div
第三屏幕
CL
HTML 5
210
215

图 2

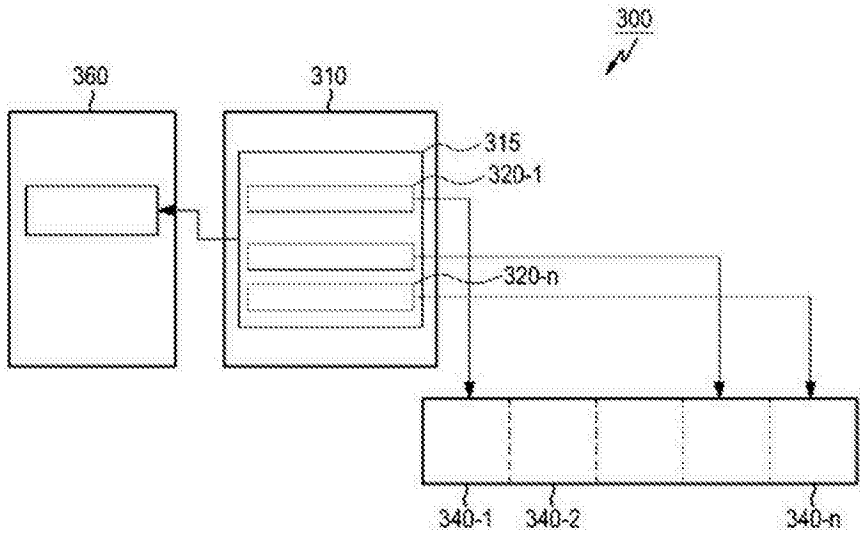

300
360
310
315
320-1
320-n
340-1
340-2
340-n

图 3

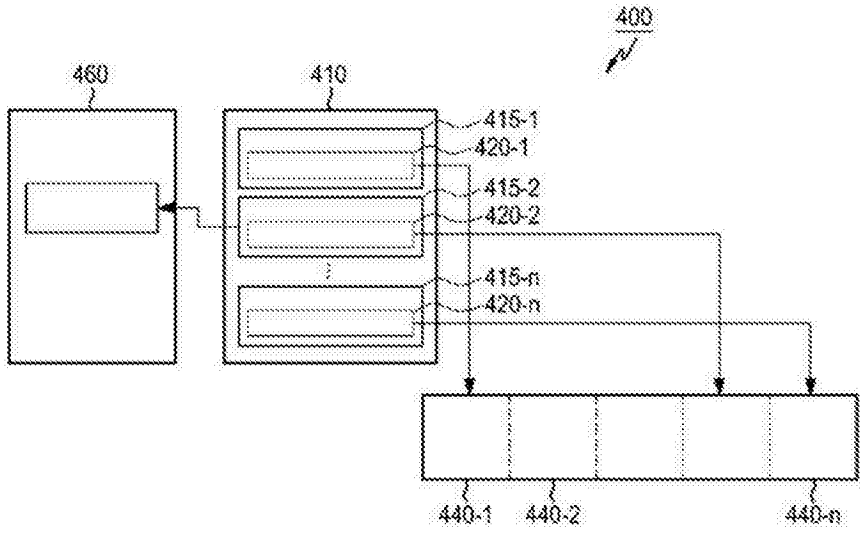

400
460
410
415-1
420-1
415-2
420-2
415-n
420-n
440-1
440-2
440-n

图 4

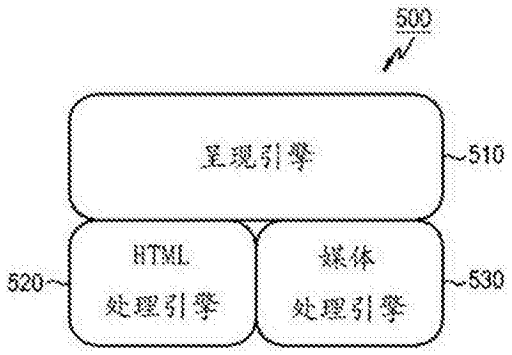

500
呈现引擎
510
HTML
处理引擎
520
媒体
处理引擎
530

图 5

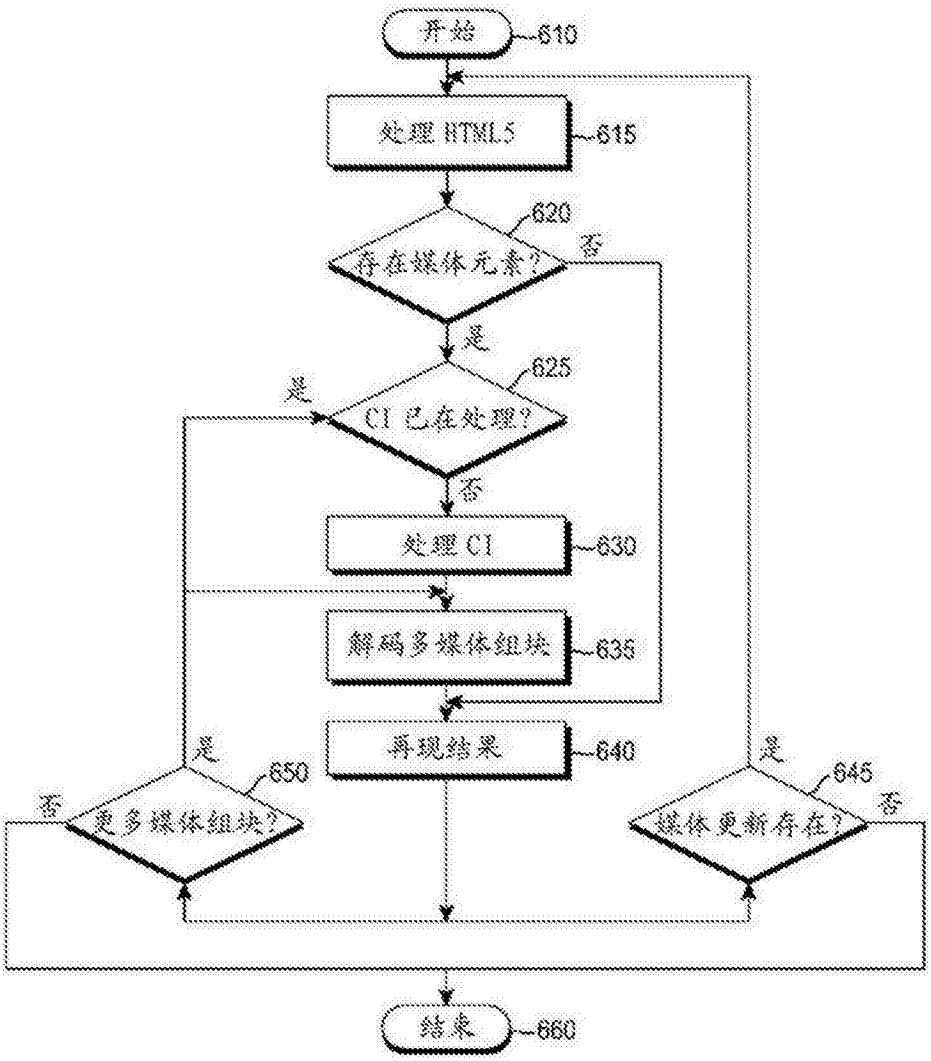

开始
610
处理 HTML5
615
620
存在媒体元素?
否
是
625
是
CI 已在处理?
否
处理 CI
630
解码多媒体组块
635
再现结果
640
是
650
否
更多媒体组块?
是
645
否
媒体更新存在?
结束
660

图 6

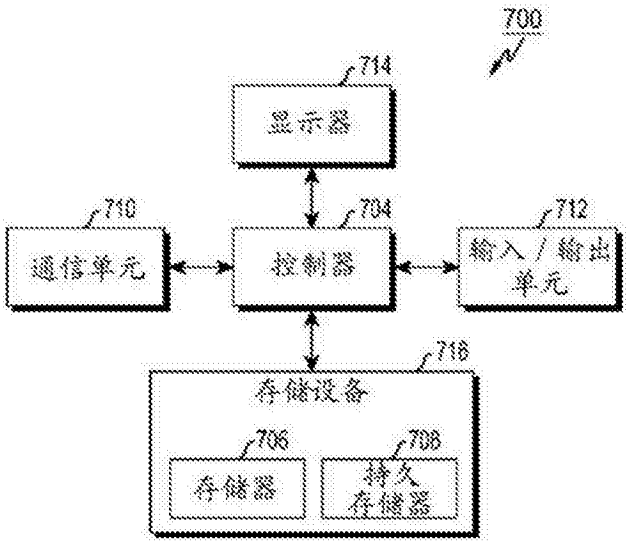

700
714
显示器
710
通信单元
704
控制器
712
输入 / 输出单元
716
存储设备
706
存储器
708
持久存储器

图 7

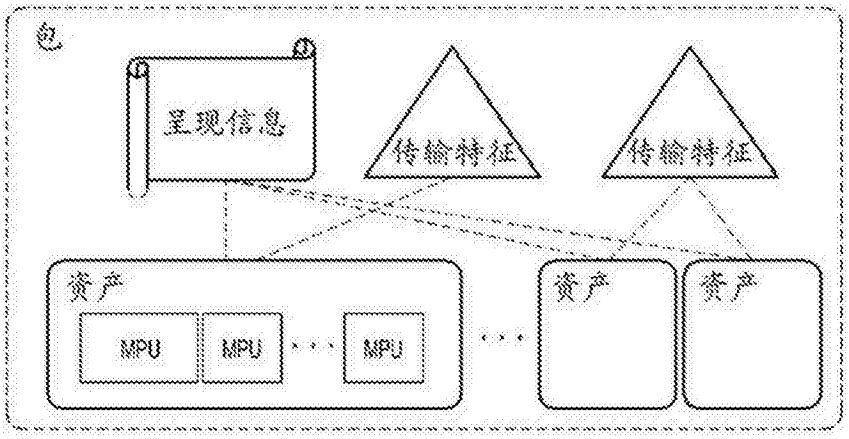

包
呈现信息
传输特征
传输特征
资产
MPU
MPU
MPU
资产
资产

图 8

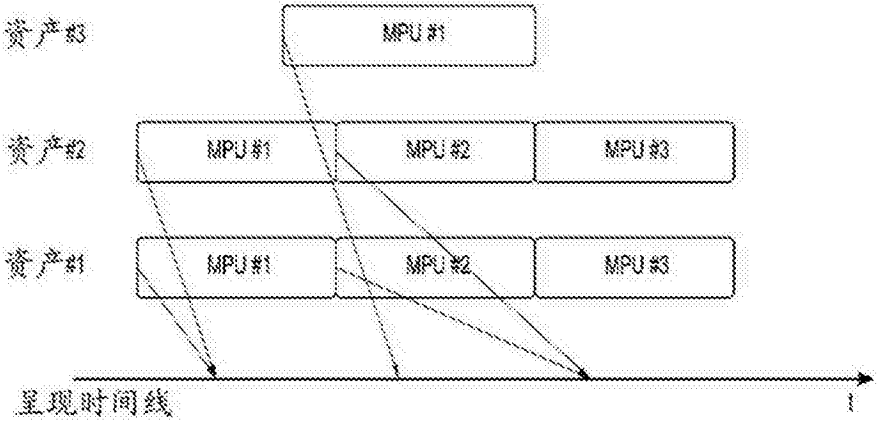

资产#3
MPU #1
资产#2
MPU #1
MPU #2
MPU #3
资产#1
MPU #1
MPU #2
MPU #3
呈现时间线
t

图 9