[19] 中华人民共和国国家知识产权局

[51] Int. Cl.
*G06F 17/30* (2006.01)

[12] 发明专利申请公布说明书

[21] 申请号 200810083308.1

[43] 公开日 2008 年 12 月 10 日

[11] 公开号 CN 101320377A

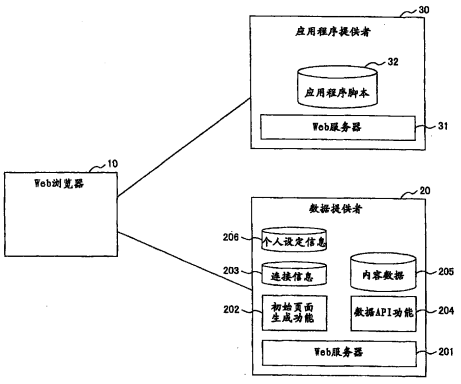
[22] 申请日 2008.3.6
[21] 申请号 200810083308.1
[30] 优先权
[32] 2007. 6. 5 [33] JP [31] 2007 - 149341
[71] 申请人 株式会社日立制作所
地址 日本东京都
[72] 发明人 小岛刚 植田良一
[74] 专利代理机构 北京银龙知识产权代理有限公司
代理人 许 静
权利要求书 4 页 说明书 17 页 附图 14 页

[54] 发明名称

# Web 应用程序的构成方法及其系统

[57] 摘要

本发明提供一种 Web 应用程序的构成方法及其系统。 可以不直接将数据传递给 Web 应用程序的提供者地利用 Web 应用程序。 提供用户界面功能的应用程序提供者(30)，将在 Web 浏览器(10)上运行的用于提供用户界面功能的应用程序脚本(32)发送至用户所操作的 Web 浏览器(10)。 使用连接信息与应用程序提供者(30)相关联、提供与数据操作相关的功能的数据提供者(20)，当需要伴随 Web 浏览器(10)上的用户操作而进行数据操作处理时，根据数据的追加・修正・删除・参照等请求来操作所保存的数据，仅将其结果信息返回给作为请求源的 Web 浏览器(10)。



1. 一种 Web 应用程序构成方法，根据来自利用 Web 浏览器的用户的请求，在 Web 上提供应用程序的服务，其特征在于，

使用对所述 Web 浏览器提供用户界面功能的用户界面服务器、和保存数据并提供与该数据的操作相关的数据操作功能的数据操作服务器，

所述 Web 浏览器，基于所述用户界面服务器的所述用户界面功能，进行画面显示，

所述数据操作服务器，根据从进行了所述画面显示的所述 Web 浏览器输入的请求，执行所述数据操作功能，将操作结果发送至所述 Web 浏览器。

2. 根据权利要求 1 所述的 Web 应用程序构成方法，其特征在于，

所述数据操作服务器，保存为与所述数据操作服务器连接而需要的连接信息，按照该连接信息来设定与所述用户界面服务器的连接方法，使所述数据操作功能和所述用户界面功能协同动作。

3. 根据权利要求 2 所述的 Web 应用程序构成方法，其特征在于，

所述数据操作服务器，生成进行按照所述连接信息来设定与所述用户界面服务器的连接方法的处理的画面信息，并提供给所述 Web 浏览器。

4. 根据权利要求 2 所述的 Web 应用程序构成方法，其特征在于，

所述数据操作服务器，还保存针对每个用户进行所述用户界面功能的动作设定的用户设定信息，生成进行按照该用户设定信息和所述连接信息来设定与所述用户界面服务器的连接方法的处理的画面信息，并提供给所述 Web 浏览器。

5. 根据权利要求 2 所述的 Web 应用程序构成方法，其特征在于，

所述用户界面服务器，保存针对每个用户进行所述用户界面功能的动作设定的用户设定信息，并按照该用户设定信息进行动作。

6. 根据权利要求 1 所述的 Web 应用程序构成方法，其特征在于，

所述数据操作服务器还具备对保存的所述数据进行分析的数据分析功能。

7. 根据权利要求 6 所述的 Web 应用程序构成方法，其特征在于，

所述数据操作服务器，将所述数据分析功能的分析结果作为所述数据操作

功能来进行提供。

8. 根据权利要求 6 所述的 Web 应用程序构成方法，其特征在于，

所述用户界面服务器的所述用户界面功能，利用所述数据操作服务器提供的所述分析结果。

9. 一种 Web 应用程序构成系统，用于根据来自用户的请求，在 Web 上提供应用程序的服务，其特征在于，

具有：

所述用户输入请求的 Web 浏览器；

提供与所述用户的请求对应的用户界面功能的用户界面服务器；以及

存储数据、构成与数据操作相关的功能的数据操作服务器，

所述 Web 浏览器，从所述用户界面服务器接收用于所述用户界面功能的应用程序脚本，使用接收到的所述应用程序脚本进行画面显示，

所述数据操作服务器，根据从所述 Web 浏览器输入的请求，执行所述数据操作功能，将操作结果发送至所述 Web 浏览器。

10. 根据权利要求 9 所述的 Web 应用程序构成系统，其特征在于，

所述数据操作服务器，对登录了为与所述数据操作服务器连接而需要的信息的连接信息进行存储，按照所述连接信息来设定与所述用户界面服务器的连接方法，并将所述数据连接功能与所述用户界面功能相关联。

11. 根据权利要求 10 所述的 Web 应用程序构成系统，其特征在于，

所述数据操作服务器，生成进行按照所述连接信息来设定与所述用户界面服务器的连接方法的处理的画面信息，并将生成的所述画面信息提供给所述 Web 浏览器。

12. 根据权利要求 11 所述的 Web 应用程序构成系统，其特征在于，

所述数据操作服务器，存储针对每个用户进行所述用户界面功能的动作设定的用户设定信息，并按照所述用户设定信息针对所述每个用户生成所述画面信息。

13. 根据权利要求 10 所述的 Web 应用程序构成系统，其特征在于，

所述用户界面服务器，存储针对每个用户进行所述用户界面功能的动作设定的用户设定信息，并按照所述用户设定信息进行动作。

14. 根据权利要求 9 所述的 Web 应用程序构成系统，其特征在于，

所述数据操作服务器存储了内容数据作为所述数据，

所述数据操作服务器还具备对存储的所述内容数据进行分析的分析功能，将所述分析功能的分析结果作为所述内容数据的一部分来存储。

15. 根据权利要求 14 所述的 Web 应用程序构成系统，其特征在于，

所述数据操作服务器将所述分析结果作为与所述数据操作相关的功能来进行提供。

16. 根据权利要求 14 所述的 Web 应用程序构成系统，其特征在于，

所述用户界面服务器存储广告信息，具有分发所述广告信息的分发功能，

所述数据操作服务器对来自所述 Web 浏览器的内容的特征信息的取得请求进行应答，将所述分析结果作为所述特征信息提供给所述 Web 浏览器，

所述 Web 浏览器，将适合于所述特征信息的广告信息的取得请求发送至所述用户界面服务器，

所述用户界面服务器的所述分发功能，将存储的所述广告信息中与所述特征信息相关联的广告信息分发至所述 Web 浏览器。

17. 根据权利要求 9 所述的 Web 应用程序构成系统，其特征在于，

将所述用户界面服务器和所述数据操作服务器设置在数据管理企业内。

18. 一种 Web 应用程序构成系统，用于根据来自用户企业的请求，在 Web 上提供应用程序的服务，其特征在于，

具有：

所述用户企业输入请求的 Web 浏览器；

设置在通过网络与所述用户企业连接的数据管理企业内、并提供与所述用户企业的请求相对应的用户界面功能的用户界面服务器；以及

设置在所述数据管理企业内、存储数据并构成与数据操作相关的功能的数据操作服务器，

所述 Web 浏览器从所述用户界面服务器接收用于所述用户界面功能的应用程序脚本，使用接收到的所述应用程序脚本来显示画面，

所述数据操作服务器，根据从所述 Web 浏览器输入的请求，执行所述数据操作功能，将操作结果发送至所述 Web 浏览器。

19. 根据权利要求 18 所述的 Web 应用程序构成系统，其特征在于，

所述数据操作服务器，对登录了为与所述数据操作服务器连接而需要的信息的连接信息进行存储，按照所述连接信息来设定与所述用户界面服务器的连接方法，并将所述数据连接功能与所述用户界面功能相关联。

20. 根据权利要求 18 所述的 Web 应用程序构成系统，其特征在于，

所述数据操作服务器，生成进行按照所述连接信息来设定与所述用户界面服务器的连接方法的处理的画面信息，并将生成的所述画面信息提供给所述 Web 浏览器。

# Web应用程序的构成方法及其系统

## 技术领域

本发明涉及 Web 应用程序的系统构成方法，涉及可以将用户（使用者）处理的数据的存储目的地限制在希望的场所的技术。

## 背景技术

目前，作为在客户机装置上执行的桌面应用程序而安装的企业内的业务系统有很多，但作为 Web 上的服务（Web 应用程序）而被提供的刚起步不久。随着这种 Web 上的服务的广泛利用，存在用户个人的隐私信息或公司内的营业信息等向 Web 服务器侧泄漏的问题。

为了应对该问题，在专利文献 1 中记载了通过存储向提供服务的 WEB 服务提供装置以及认证机构发送各个信息时进行的加密、解密中所需要的用户密钥的防篡改性存储装置、以及具备通过该防篡改性存储装置中存储的用户密钥进行加密处理的 CPU 的装置，来保护用户的隐私信息的方法。

通过该方法可以减少对用户而言烦杂的处理操作，可以减小由第三者获得密钥等用户隐私信息的危险性。

另外，在专利文献 2 中记载了如下方法：当接收到将所选择的关键字以及所选择的要素分别保密而发送的希望购入信息时，根据希望购入信息以及密钥来计算发送密钥并发送至用户终端，在用户终端通过根据送付密钥、所选择的要素、多个信息，仅使与选择的关键字所对应的信息解密的装置，在使用户选择的信息（其摘要）对信息提供者保密的状态下，向用户提供所选择的信息的详细内容。

通过该方法，用户可以将需要何种信息对信息提供者保密地、从信息提供者获得想取得的信息。

【专利文献 1】特开 2003－304234 号公报

【专利文献 2】特开 2006－39674 号公报

在现有技术中，存在必须在用户一侧的装置和信息提供者一侧的装置中设

置用于信息保密的特别装置和程序的问题。另外存在如下问题：信息保密的对象是在认证处理时被通信的信息、或者是表示想要取得的信息的信息，以限定的情况或信息作为对象，无法通用。

发明内容

鉴于这些问题而提出本发明，本发明提供一种系统，该系统可以不在用户一侧新引入特别的功能或装置、并且不将用户的数据传递给应用程序提供者地利用提供者所提供的应用程序。

为达到上述目的，本发明的系统构成为，将通常由单一服务器（组织）提供的、在 Web 上服务的应用程序（以下称为 Web 应用程序）的功能划分为两部分，通过提供用户界面的功能的用户界面服务器（组织）（以下称为应用程序提供者）和、提供与数据操作相关的功能的数据操作服务器（组织）（以下称为数据提供者）的两个服务器（组织）来提供 Web 应用程序的功能。

在以往的 Web 应用程序中，用户界面的功能由通过提供 Web 应用程序的服务器生成的 HTML 构成，重复进行以下处理：当用户输入信息时，将该信息发送至提供 Web 应用程序的服务器，Web 应用程序服务器生成反映所输入的信息的 HTML 并将其发送至 Web 浏览器，显示新的画面。与数据操作相关的功能也由同一服务器提供，当生成反映上述用户的输入的 HTML 时，也执行与数据操作相关的处理。

本发明的系统中，应用程序提供者将 Web 浏览器上运行的程序（以下称为应用程序）提供给用户操作的 Web 浏览器。该应用程序是在 Web 浏览器上运行的程序，因此当变更画面上的显示时，可以不需要访问应用程序提供者的服务器。

本发明的系统中，数据提供者不进行用于提供用户界面的 HTML 的生成处理，而根据数据的追加·修正·删除·参照等请求对数据提供者内的服务器内保存的数据进行操作，仅将其结果信息返回请求源。

在应用程序中需要伴随用户操作来进行数据操作处理的部分，按照数据提供者的通信规范，对数据提供者进行通信，由此可以使数据提供者进行数据操作处理。

即，在本发明中，提供以下根据来自利用 Web 浏览器的用户的请求，在

Web 上提供应用程序的服务的 Web 应用程序构成方法及其构成系统，具备对 Web 浏览器提供用户界面功能的用户界面服务器；以及保存数据，提供与该数据的操作相关的数据操作功能的数据操作服务器，Web 浏览器从用户界面服务器接收用于用户界面功能的应用程序（应用程序脚本），基于接收到的应用程序（应用程序脚本）来显示画面，根据从该进行画面显示的 Web 浏览器输入的请求，数据操作服务器执行数据操作功能，Web 浏览器接收其操作结果，由此更新画面显示。

根据本发明，可以向用户提供如下系统，该系统可以将 Web 应用程序的功能划分为用户界面的功能、与数据操作相关的功能这两部分，分别配置在不同的服务器上，可以不将 Web 应用程序所处理的数据直接传递给提供用户界面的功能的用户界面服务器地利用 Web 应用程序。

而且，将提供与数据操作相关的功能的数据操作服务器适当地配置在用户的组织内或用户信赖的组织内，由此可以不将用户处理的信息泄漏到外部地利用 Web 应用程序，该 Web 应用程序利用了用户的组织以及作为用户信赖的组织的外部组织所提供的用户界面的功能。

而且，可以容易地提供通过多个不同的用户界面功能对公共数据进行操作的系统。而且，由于还可以将提供多个用户界面功能的服务器作为利用对象而登录并可以选择，因此可以排除将数据发送至外部等非法的程序。

附图说明

图 1 是表示实施例 1 的系统的功能结构和网络结构的图。

图 2 是表示实施例 1 的初始页面的一例的图。

图 3 是表示实施例 1 的系统的处理流程的图。

图 4 是表示实施例 1 中的连接信息所保存的应用程序脚本信息的图。

图 5 是表示实施例 1 中的连接信息所保存的数据 API 信息的一例的图。

图 6 是表示实施例 1 中的个人设定信息 206 的一例的图。

图 7 是表示实施例 2 中的本发明的系统的处理流程的图。

图 8 是表示实施例 3 中的连接信息所保存的数据 API 信息的一例的图。

图 9 是表示实施例 4 中的系统的处理流程的图。

图 10 是表示实施例 1 中的数据 API 的通信形式的图。

图 11 是表示实施例 1 中的预定管理应用程序所提供的画面的一例的图。

图 12 是表示实施例 2 中的应用程序脚本选择页面的一例的图。

图 13 是表示实施例 1 中的追加数据提供者的信息的一例的图。

图 14 是表示实施例 5 中的系统的功能结构和网络结构的图。

图 15 是表示实施例 5 中的预定管理应用程序所提供的画面的一例的图。

图 16 是表示实施例 5 中的系统的处理流程的图。

图 17 是表示实施例 5 中的广告信息 34 的一例的图。

图 18 是表示实施例 1 中的应用例的网络结构的图。

符号说明

10 Web 浏览器；20 数据提供者；30、30A、30B、30C、30D 应用程序提供者；31 Web 服务器；32 应用程序脚本；33 广告分发功能；34 广告信息；201 Web 服务器；202 初始页面生成功能；203 连接信息；204 数据 API 功能；205 内容数据；206 个人设定信息；207 数据分析功能；601～603 个人设定信息的记录；1701～1703 特征信息与应分发的广告画面的 URL 的对应关系

具体实施方式

以下，参照附图说明本发明的实施方式。

【实施例 1】

首先，通过图 1 对第一实施例的网络系统的整体结构进行说明。

实施例 1 的网络系统中包含与网络连接的多个信息处理装置 10、20、30。这些信息处理装置中包含有：（1）安装了 Web 浏览器的 1 台以上的信息处理装置（以下称为 Web 浏览器）10；（2）对本系统中所提供的 Web 应用程序的功能中的、与数据的检索·追加·删除·编辑·参照等数据操作相关的功能进行提供的信息处理装置（以下称为数据提供者）20；（3）对本系统中所提供的 Web 应用程序的功能中的、在 Web 浏览器上提供的用户界面（user interface）的功能进行提供的信息处理装置（以下称为应用程序提供者）30。此外，图 1 中存在与用户数相等的 Web 浏览器 10，但在此为了说明仅表示了一个。另外，多个信息处理装置分别由通常的计算机系统构成，当然具有处理部（CPU）、存储器或磁盘装置等存储部、与外部网络连接的通信部等。

以下，对实施例 1 的构成要素的功能以及处理步骤进行说明。数据提供者

20 由以下各部构成：Web 服务器 201，通过网络接受来自 Web 浏览器 10 的处理请求，返回与处理请求相对应的 Web 页面；初始页面生成功能 202，生成用于执行本系统的 Web 应用程序的初始页面；连接信息 203，保存用于将作为应用程序提供者 30 提供的应用程序的应用程序脚本 32、和数据提供者 20 提供的功能进行连接所需的信息；数据 API（Application Program Interface）功能 204，提供用于对本系统的 Web 应用程序中处理的数据进行操作处理的用户界面功能；本系统的 Web 应用中处理的内容数据 205；以及用户的个人设定信息 206。连接信息 203、内容数据 205、个人设定信息 206 被存储在存储部中，初始页面生成功能 202 和数据 API 功能 204 是程序，通常被存储在上述存储部中，在处理部中被执行。

另一方面，应用程序提供者 30 由：通过网络接受来自 Web 浏览器 10 的处理请求，返回与处理请求对应的应用程序脚本的 Web 服务器 31；和提供本系统的 Web 应用程序的功能中的用户界面功能的应用程序、即应用程序脚本 32 构成。该应用程序脚本 32 等程序通常被存储在构成应用程序提供者 30 的信息处理装置的存储部中。

实施例 1 的系统，在 Web 浏览器 10 和数据提供者 20 以及应用程序提供者 30 之间通过网络进行通信，同时有关数据的操作处理（数据的生成、编辑、参照、删除），利用数据提供者 20 的数据 API 功能，关于用户界面功能，向用户提供利用应用程序提供者 30 提供的功能的应用程序执行环境。

以下，具体说明实施例 1 的系统的处理步骤。

在此，作为本系统的应用例而举例说明预定管理应用程序。图 11 中表示预定管理应用程序所提供的画面的一例。在该图中，1101 的部分是用于新追加从 2007 年 2 月 21 日开始的预定的部分，是用户已输入了预定信息的状态，通过按下 1102 的 add 按钮，可以追加预定信息。1103 的部分是显示已经登录的预定信息的部分，表示在 2007 年 2 月 23 日有“Web meeting”的预定。例中的预定管理应用程序在画面上显示日历，并且在该日历上显示用户的预定信息，用户可以进行预定信息的追加·修正·删除·阅览的操作。

在实施例 1 的系统中应用预定管理应用程序时，应用程序提供者 30 提供应用程序脚本 32，该应用程序脚本 32 提供在 Web 浏览器 10 的画面上显示日

历和预定信息、或者进行预定信息的操作处理的功能，数据提供者 20 在内容数据 205 中保存预定信息，通过数据 API 功能 204 提供预定信息的追加·修正·删除·参照的功能。

将应用程序提供者 30 提供的应用程序脚本 32 记述为：关于其数据操作处理，按照数据提供者 20 提供的数据 API 功能 204 对数据提供者 20 进行通信，在数据提供者 20 中进行数据操作处理。换言之，在本系统中，以数据 API 作为中介，应用程序提供者 30 和数据提供者 20 协同动作。

图 2 是表示数据提供者 20 的初始页面生成功能 202 所生成的初始页面的 HTML（HyperText Markup Language）的一例的图。其中，为了说明而在左端标注了行号。

图 3 是表示实施例 1 的系统的处理步骤的流程图。以下，按照图 3 的流程说明实施例 1 的处理步骤。此外，在图 3 中，步骤 304 以后的虚线包围的部分，是在第二实施例以后的实施例中也共同使用的本系统中的基本处理流程部分。

首先，当利用本系统时，用户对 Web 浏览器 10 进行访问数据提供者 20 提供的初始页面的操作（步骤 301）。Web 浏览器 10 通过用户操作，将初始页面的取得请求发送至数据提供者 20（步骤 302）。数据提供者 20 当通过 Web 服务器 201 接收到初始页面的取得请求时，生成在图 2 中表示其一例的初始页面，将生成的初始页面返回至作为请求源的 Web 浏览器 10（步骤 303）。

初始页面生成功能 202 按照在连接信息 203 中设定的信息来生成初始页面。图 4、图 5 是该连接信息 203 的一个具体例子，图 4 是应用程序提供者 30 提供的应用程序脚本的 URL 的信息（以下称为应用程序脚本信息）的一例，图 5 是数据提供者 20 提供的数据 API 功能 204 与 URL 的对应信息（以下称为数据 API 信息）的一例。另外，图 6 表示个人设定信息 206 的一例。

初始页面基本上如图 2 的一例那样，由如下两部分构成：第 4 行由虚线包围的、应用程序提供者 30 提供的应用程序脚本 32 的参照记述部分；以及从第 5 行到第 14 行由虚线包围的、用于对应用程序脚本反映数据提供者 20 提供的数据 API 功能 204 与 URL 的对应信息的设定、和用户的个人设定的脚本记述部分。

第一部分、即应用程序脚本 32 的参照记述部分，根据连接信息 203 中设

定的应用程序脚本信息而生成。在图 4 的应用程序脚本信息的一例的情况下，例如，若用户利用的应用程序脚本的名称为 calendar app A，则在应用程序脚本的参照记述中使用 http://appproviderA.com/script/calendar.js。

第二部分、即对应用程序脚本反映由数据提供者 20 提供的数据 API 功能与 URL 的对应信息的设定、和用户的个人设定的脚本记述部分，根据连接信息 203 中设定的数据 API 信息、和个人设定信息 206 的信息而生成。在图 5 的数据 API 信息的一例的情况下，若提供的数据 API 的名称为“GET”，则将“http://dataprovider.com/calendar/api/get/$user$”用作对应用程序脚本设定的 URL；若提供的数据 API 的名称为“ADD”，则将“http://dataprovider.com/calendar/api/add/$user$”用作对应用程序脚本设定的 URL；若提供的数据 API 的名称为“EDIT”，则将“http://dataprovider.com/calendar/api/edit/$user$/$id$”用作对应用程序脚本设定的 URL。

此外，在上述例子中的 URL 中，“$user$”和“$id$”的部分表示在脚本的处理中动态变化的部分，表示分别将操作对象数据所对应的用户 ID、预定信息 ID 填入该部分来利用。例如，当想要对用户 ID 为“kojima”的数据进行处理时，数据 API“GET”的 URL 成为“http://dataprovider.com/calendar/api/get/kojima”。另外，图 5 中“ANALYZE”与实施例 1 无关，因此在此省略说明。

在如图 6 的例子那样设定了个人设定信息 206 的情况下，当用户是“kojima”时，作为针对应用程序脚本的个人设定信息，信息类别 initView 的值使用 monthly 作为日历的初始显示方法的设定信息（记录 601），信息类别 initDay 的值使用 Sunday 作为最初显示的星期的设定信息（记录 602），信息类别 lang 的值使用 ja 作为在显示中利用的语言的设定信息（记录 603）。

通过以上的步骤，数据提供者 20 的初始页面生成功能 202 生成初始页面。Web 浏览器 10 当接收到如此生成的初始页面时，在画面上显示作为初始页面而接收到的 HTML（图 3 的步骤 304）。接收到的初始页面中如图 2 的例子那样，包含对应用程序脚本 32 的参照记述，因此将该参照记述中记载的 URL 的脚本的取得请求发送至应用程序提供者 30。应用程序提供者 30，当通过 Web 服务器 31 接收到脚本的取得请求时，将应用程序提供者 30 所保存的脚本中按

照取得请求的应用程序脚本32返回给作为请求源的Web浏览器10(步骤305)。

接收到应用程序脚本 32 的 Web 浏览器 10 开始执行接收到的应用程序脚本 32(步骤 306)。接着,按照初始页面内记述的脚本,进行数据提供者 20 提供的数据 API 的 URL 设定处理(步骤 307)。在来自应用程序提供者 30 的脚本 32 内定义该设定处理中所使用的函数。

在图 2 的例子中,以第 8 行到第 10 行中定义的数据 API 与 URL 的关系信息、以及个人设定信息 206 中设定的信息作为自变量,在第 11 行的 2001 的位置执行 executeCalendar 函数,而该 executeCalendar 函数在应用程序提供者 30 提供的脚本 32 内被定义,为了取得在 2002 的位置表示的作为自变量而传递的数据 API 与 URL 的关系信息,在数据操作处理时利用而运行。另外,按照在 2003 的位置表示的第二个自变量的值来控制用户界面的动作。在此,例如作为用户的个人设定的值,在个人设定信息 206 中如图 6 的一例那样进行了设定,应用程序脚本在日历的初始显示时进行月份显示,使最初显示的星期为星期日,在所显示的语言中使用日本语。

在步骤 306 中开始脚本的执行,但在步骤 307 的数据 API 的 URL 的设定处理后开始进行伴随用户的操作来执行数据操作的处理。

步骤 306 后,通过应用程序脚本 32 的执行,Web 浏览器 10 在画面上显示信息,或者显示用于操作数据的用户界面。在预定管理应用程序的情况下,显示日历,显示用于追加与所显示的日期对应的预定信息、或者用于编辑或删除在所显示的日期设定的预定信息的界面。

接着,如图 3 的处理流程所示,当用户对 Web 浏览器 10 进行追加与所显示的日期对应的预定信息的操作时(步骤 308),Web 浏览器 10 使该操作所对应的事件发生。应用程序脚本 32 以该事件的发生为契机,根据用户输入的信息进行预定信息的追加处理。按照用户输入的信息,生成按照数据提供者提供的数据 API 形式的请求信息,通过异步通信将该请求信息发送至数据提供者 20(步骤 309)。

通过 Web 服务器 201 接收到请求信息的数据提供者 20,按照该请求信息,通过数据 API 功能 204 对内容数据 205 进行追加处理,将追加处理结果返回给作为请求源的 Web 浏览器 10(步骤 310)。

接收到追加处理结果的 Web 浏览器 10，将追加处理结果的信息传递到进行了追加请求发送处理的应用程序脚本 32 的相应处理部。应用程序脚本 32 按照追加处理结果的信息，进行将该意思显示在画面上等针对追加处理结果的处理（步骤 311）。

图 10 是表示实施例 1 中的数据 API 的通信形式的一例的表。应用程序脚本 32 在步骤 309 中按照图 10 的通信形式向数据提供者 20 发送请求信息。另外，在步骤 311 中，分析从请求信息的发送结果中得到的信息，进行结果的画面显示处理。在图 10 的 1001 中意味着，在预定信息的追加处理时，当想要追加将预定的开始日设为 2007 年 2 月 21 日、将结束日设为 2007 年 2 月 22 日、将标题设为 “web conference”、将场所设为 “tokyo” 的预定时，生成 “startdate=20070221&enddate=20070222&title=web+conference&place=tokyo” 的请求参数，其结果意味着当完成了追加时由数据提供者 20 返回“( {”status”:”ok”} )”。另外，1002 意味着，在预定信息的检索处理时，以开始日在 2007 年 1 月 1 日以后、在标题或场所中包含 “web” 字符串为条件来进行检索时，生成 “startdate=20070101&keyword=web” 的请求参数，其结果意味着，作为符合条件的预定信息，由数据提供者 20 返回 “({”schedules”:[{”startdate”:”20070220”,”enddate”:”20070222”,”title”:”web conference”,”place”:”tokyo”},{”startdate”:”20070223”,”enddate”:”20070223”,”title”:”web meeting”,”place”:”meeting room”},……]})”。

同样地，在图 3 的处理流程中，当用户对 Web 浏览器 10 进行修正所显示的预定信息的操作时（步骤 312），Web 浏览器 10 使该操作所对应的事件发生。应用程序脚本 32 以该事件的发生为契机，根据用户输入的信息进行预定信息的修正处理。按照用户输入的信息，生成按照数据提供者提供的数据 API 的形式的请求信息，将该请求信息发送至数据提供者 20（步骤 313）。

通过 Web 服务器 201 接收到请求信息的数据提供者 20，按照该请求信息，通过数据 API 功能 204 对内容数据 205 进行修正处理，将修正处理结果返回给作为请求源的 Web 浏览器 10（步骤 314）。

接收到修正处理结果的 Web 浏览器 10，将修正处理结果的信息传递至进行了修正请求发送处理的应用程序脚本 32 的相应处理部。应用程序脚本 32，

按照修正处理结果的信息，进行将该意思显示在画面上等针对修正处理的结果的处理（步骤 315）。

关于其它删除或参照预定信息的用户操作也相同，Web 浏览器 10 对数据提供者 20 发送对应的请求信息，接收到该请求信息的数据提供者 20 执行数据操作处理。

此外，步骤 308～311 和步骤 312～步骤 315 在说明上按照该顺序来描述，但也可以是以用户的操作为契机而开始的步骤，与顺序无关地进行处理。

根据以上说明的实施例 1 的系统，当其用户利用 Web 应用程序时，可以不将其中所处理的数据直接传递至 Web 应用程序中提供用户界面功能的应用程序提供者 30 而进行利用。另外，Web 应用程序的开发者，通过以数据 API 的形式为中介，可以将 Web 应用程序的用户界面功能和数据操作处理功能分离，可以将它们分别配置在不同的域的服务器上。

由此，企业内系统的开发者，通过将因特网上公开的应用程序功能所对应的数据提供者 20 设置在本公司内，可以将数据限制在本公司内来利用在因特网上公开的应用程序功能。反之，因特网上的应用程序开发者，通过开发与设置在企业内的数据提供者所对应的数据 API 相对应的应用程序功能，可以不直接处理企业内的数据，而在因特网上配置应用程序并进行服务。此外，也可以将个人设定信息 206 配置在应用程序提供者 20 中。

另外，在本实施例的系统结构中，可以将数据提供者 20 设置在与利用 Web 浏览器 10 的企业不同的网络中来使其工作。因此也可以采取在可以更安全地保护数据的数据中心等中配置数据提供者 20 来利用的形态。因此，如图 18 所示，可以安全地保护顾客企业的数据，同时也可以应用于可利用外部应用程序的数据管理业务中。

图 18 中表示，30A、30B、30C、30D 都与图 1 的应用程序提供者 30 相同，具有 Web 服务器 31 和应用程序脚本 32，分别由不同企业来提供的情况。表示了由数据管理企业提供应用程序提供者 30D 和数据提供者 20，该数据管理企业进行同时进行应用程序的提供服务和数据管理的服务的业务。数据提供者 20 具有与图 1 的数据提供者相同的结构。由此，作为 Web 浏览器 10 的用户的企业，可以实现利用数据管理企业所提供的应用程序，同时也保证了数据的安

全管理。

【实施例2】

在上述的实施例1中，说明了将Web应用程序的功能中的用户界面的功能和数据操作的处理功能分离的构造，而通过从数据提供者20提供的初始页面来访问的应用程序脚本对应于一种。在数据提供者提供的数据操作功能中，提供多个应用程序提供者所对应的功能时，用户可以选择利用对数据提供者提供的数据操作功能进行处理的用户界面功能。通过使实施例1中说明的初始页面生成功能202具备应用程序提供者的选择页面生成功能，可以提供该功能。

以下，对于实施例1中说明的系统，使用图7的流程图说明还具备应用程序提供者的选择处理的第二实施例的系统的处理步骤。

首先，用户对Web浏览器10进行访问由数据提供者20提供的应用程序脚本选择页面（以下称为应用程序选择页面）的操作（步骤701）。此外，应用程序脚本选择页面成为具备用户界面的结构，该用户界面对利用数据提供者20提供的数据操作功能所对应的应用程序脚本中的哪个应用程序脚本进行选择。图12是应用程序脚本选择页面的一例，成为通过按下按钮可以选择该按钮上所记载的名称的应用程序脚本的结构。

Web浏览器10通过用户操作，向数据提供者20发送应用程序选择页面的取得请求（步骤702）。数据提供者20，当通过Web服务器201接收到应用程序选择页面的取得请求时，生成应用程序选择页面，将生成的应用程序选择页面返回给作为请求源的Web浏览器10（步骤703）。

应用程序选择页面，根据连接信息203中保存的、由数据提供者20提供的数据API所对应的应用程序脚本的信息（以下称为脚本对应信息），生成初始页面生成功能202。图4是应用程序脚本信息的一例。当数据提供者20提供的数据API的名称为“calendar API”时，意味着存在名称为“calendar app A”和“Bcalendar”的应用程序脚本。因此，初始页面生成功能202生成作为可以选择的应用程序脚本而存在“calendar app A”和“Bcalendar”、让用户选择某一个的应用程序选择页面。

接收到应用程序选择页面的Web浏览器10，在画面上显示应用程序选择页面（步骤704）。

用户阅览在画面上显示的应用程序脚本的信息，对 Web 浏览器 10 进行选择某个应用程序脚本的操作（705）。当用户进行应用程序脚本的选择操作时，Web 浏览器 10 将所选择的应用程序脚本的信息、和包含对所选择的应用程序脚本的参照记述的初始页面取得请求一起，发送至数据提供者 20（步骤 706）。数据提供者 20，当通过 Web 服务器 201 接收到针对所选择的应用程序脚本的初始页面取得请求时，通过初始页面生成处理功能 202 生成针对所选择的应用程序脚本的初始页面，将生成的初始页面返回给作为请求源的 Web 浏览器 10（步骤 706）。此时的初始页面的生成处理与实施例 1 中说明的图 3 的步骤 303 相同。

接收到初始页面的 Web 浏览器 10 进行初始页面的显示处理（步骤 304）。以后的处理流程与实施例 1 中说明的图 3 的步骤 304 以后的基本流程相同。

根据本实施例，可以向用户提供如下系统，该系统可以选择执行由同一数据提供者 20 提供的数据 API 所对应的多个应用程序脚本。另外，无论此时利用哪个应用程序脚本，所处理的数据都存在于数据提供者 20 中，因此可以通过多个应用程序脚本操作公共的数据。

【实施例 3】

在实施例 2 中说明了选择利用多个应用程序脚本 32 的功能，但也可以构成可对多个数据提供者提供的信息进行处理的系统。若是预定管理系统的例子，则可以在本公司内的数据提供者中管理与工作相关的预定信息，从外部的数据提供者取得节日信息或天气预报信息，将其与工作相关的预定信息并列显示。

在实施例 1 中，作为连接信息 203 的一例而表示了图 5，并描述了如下方法：为包含按照该信息将数据提供者所提供的数据 API 的信息设定在应用程序提供者 30 所提供的应用程序脚本 32 中的处理而生成初始页面，由此，应用程序脚本 32 在进行数据操作处理时，可以按照数据 API 的信息进行通信处理。与之相同，将具有其它想要取得的信息的数据提供者的信息保存在连接信息 203 中，为包含将该信息设定在应用程序脚本 32 中的处理而生成初始页面，由此，应用程序脚本 32 在进行取得其它数据提供者所提供的信息的处理时，可以按照该信息进行通信处理。

将这种情况作为第三实施例进行说明。此时，应用程序脚本 32 除了在实施例 1 中说明的预定信息管理应用程序的用户界面功能以外，还具有取得其它数据提供者所提供的信息、进行以该信息的种类所对应的形式将该信息与预定信息并列显示的处理的功能，具备设定与其它数据提供者的通信所需的信息的单元。

图 13 是在连接信息 203 中设定的、具有其它想要取得的信息的数据提供者的信息的一例。在实施例 1 中说明的处理流程中，在图 3 的步骤 303 中生成初始页面时，按照图 13 那样的信息，追加对具有其它想要取得的信息的数据提供者的信息进行设定的脚本记述（以下称为追加数据提供者设定）。图 13 的 1301 表示信息的种类为“holiday”、URL 为“http://www.holidaycalendar.com/api/jsonp”的数据提供者的信息为取得候补，1302 表示信息的种类为“weather”、URL 为“http://www.weatherinformation.com/jsonpapi”的数据提供者的情况。

设定与应用程序脚本 32 所具有的其它数据提供者的通信所需要的信息的函数的名称为“setOptionnalInformationURL”时，追加生成“setOptionnalInformationURL({”type”:”holiday”,”url”:”http://www.holidaycalendar.com/api/jsonp”},{”type”:”weather”,”url”:”http://www.weatherinformation.com/jsonpapi”});”的记述。图 2 的初始页面的一例中，当追加了该处理时，将该记述插入第 11 行和第 12 行之间。

根据本实施例 3，应用程序脚本 32 可以利用多个数据提供者所提供的信息，可以构建显示多种信息的应用程序。此外，在此对于其它数据提供者，仅以数据取得操作为例，但并不限定于数据取得操作，也可以是数据的追加或编辑、删除等更新处理。

【实施例 4】

在上述各实施例中，当用户利用本系统时，首先从操作 Web 浏览器 10 来访问数据提供者 20 开始，但也可以在搜索到想要利用的应用程序脚本后进行利用。将这种情况作为第四实施例进行说明。

将设定向应用程序脚本的连接信息的功能配置在数据提供者 20 的初始页面生成功能 202 中，由应用程序提供者提供利用该功能来设定连接信息 203

的页面（以下称为应用程序利用开始页面），由此可以向用户提供如下利用流程：搜索应用程序脚本，当搜索结果为发现了想要利用的应用程序脚本时，将其设定在数据提供者中并利用。

以下，对于实施例1中说明的系统，使用图9的流程图对具备设定向应用程序脚本的连接信息的功能的系统的处理步骤进行说明。此外，图9是在上文中从用户发现想要利用的应用程序脚本、开始显示应用程序利用开始页面的操作起开始的流程图。

首先，用户对Web浏览器10进行访问由应用程序提供者30提供的应用程序利用开始页面的操作（步骤901）。

Web浏览器10通过用户操作，将应用程序利用开始页面的取得请求发送至应用程序提供者30（步骤902）。应用程序提供者30当通过Web服务器31接收到应用程序利用开始页面的取得请求时，将应用程序利用开始页面返回给作为请求源的Web浏览器10（步骤903）。

应用程序利用开始页面具有用户界面，该用户界面使用户输入用于访问对数据提供者20的应用程序脚本的连接信息进行设定的功能的URL（以下称为连接信息设定用URL），对于所输入的URL发送该应用程序脚本的连接信息，将应用程序脚本设定在数据提供者中，是包含生成用于开始应用程序脚本的利用的初始画面的功能的页面。

接收到应用程序利用开始页面的Web浏览器10，在画面上显示应用程序利用开始页面（步骤904）。

当用户输入连接信息设定用URL时（步骤905），Web浏览器10对于所输入的连接信息设定用URL，将应用程序脚本的信息、与应用程序脚本的连接设定以及针对该应用程序脚本的初始页面生成请求一起，发送至数据提供者20（与用户输入的连接信息设定用URL对应的数据提供者20）（步骤906）。数据提供者20，当通过Web服务器201接收到连接设定以及初始页面取得请求时，通过初始页面生成处理功能202，对连接信息203追加设定应用程序脚本的信息，生成与追加设定的应用程序脚本对应的初始页面，将生成的初始页面返回给作为请求源的Web浏览器10（步骤907）。此时的初始页面的生成处理与实施例1中说明的图3的步骤303相同。

接收到初始页面的 Web 浏览器 10 进行初始页面的显示处理（步骤 304）。以后的处理流程与实施例 1 中说明的图 3 的步骤 304 以后的基本流程相同。

根据本实施方式，可以向用户提供如下利用流程：用户操作 Web 浏览器 10 在网络内搜索应用程序脚本，当搜索的结果为发现了想要利用的应用程序脚本时，将其设定在数据提供者中并利用。

【实施例 5】

在实施例 1 到实施例 4 中说明了，在应用程序的功能中，将与数据操作相关的处理配置在数据提供者中，将与用户界面相关的处理配置在应用程序提供者中的情况，但与数据操作相关的处理中，数据 API 功能以外的、用于检索处理的高速化的索引生成处理、或用于与内容数据内容相对应的广告分发的内容数据内容的特征提取处理等数据分析处理功能，也需要配置在数据提供者 20 中。

以下，对于实施例 1 中表示的预定信息管理应用程序的系统例，将还进行在画面上显示与内容数据内容相对应的广告的处理的系统作为第五实施例来说明。

图 14 是实施例 5 中的系统结构图。作为与图 1 不同的结构，对数据提供者 20 追加了数据分析功能 207，对应用程序提供者 30 追加了广告分发功能 33 以及广告信息 34。此外，数据 API 功能 204 新具备提供与通过数据分析功能 207 提取出的内容数据对应的特征信息的功能。该新具备的功能，作为图 5 的数据 API 信息的最后第 4 行中记录的 API 方法（method）、即 ANALYZE 功能而被提供，将其 URL 设为 http://dataprovider.com/calendar/api/analyze/$user$。应用程序脚本 32 通过对该 URL 进行访问，可以取得与通过数据分析功能 207 提取出的内容数据对应的特征信息。

另外，应用程序脚本 32 新具备：从数据 API 功能 204 取得与内容数据对应的特征信息，将所取得的特征信息发送至广告分发功能 33，取得关联的广告信息并在画面上显示的处理功能。

该数据分析功能 207 是上述的数据 API 功能以外的、进行用于检索处理的高速化的索引生成处理、或用于与内容数据内容对应的广告分发的内容数据内容的特征提取处理等数据分析处理的功能。在此，数据分析功能 207 对预先由

内容数据 205 保存的内容进行分析，将各内容数据内容的特征信息作为内容数据的一部分来记录。

另一方面，广告分发功能 33 是分发与内容数据内容对应的广告的功能，是当 Web 服务器 31 接收到成为内容数据内容的特征的关键字时，从广告信息 34 中提取与该关键字对应的、关联的广告，向请求源返回广告信息的功能。

图 15 是在 Web 浏览器 10 的画面上显示通过应用程序提供者 30 的广告分发功能 33 发送的广告信息的一个具体例，1501 是其广告的一例。

图 17 是广告信息 34 的一例，表示特征信息与应分发的广告图像的 URL 的对应关系。在图 17 的 1701 中表示当特征信息为 web 时，将 http://www.add.com/image/001.png 设定为与特征信息关联的、应分发的广告图像；在 1702 中表示当特征信息是 network 时，将 http://www.add.com/image/002.png 设定为与特征信息关联的、应分发的广告图像；在 1703 中表示当特征信息是 application 时，将 http://www.add.com/image/003.png 设定为与特征信息关联的、应分发的广告图像。进行广告分发的预定信息管理应用程序的处理流程，除了与广告分发相关的部分以外，与实施例 1 中说明的图 3 的预定信息管理应用程序的处理流程没用太大差别。

图 16 是在实施例 5 的系统中，接着图 3 的步骤 307 进行的用于广告分发的处理流程。以下，使用图 16 说明用于广告分发的处理流程。

开始应用程序脚本 32 的执行（步骤 306），在数据 API 信息的设定处理完成（步骤 307）后，为了取得显示中的内容的特征信息，Web 浏览器 10 根据用户的指示，对数据提供者 20 发送内容的特征信息的取得请求（步骤 1601）。数据 API 功能 204，当通过 Web 服务器 201 接收到特征信息的取得请求时，从内容数据 205 中取得与通过取得请求所指定的内容相对应的特征信息，将所取得的特征信息返回给作为请求源的 Web 浏览器 10（步骤 1602）。接收到特征信息的 Web 浏览器 10，将该信息传递至进行了特征信息的取得请求发送处理的应用程序脚本 32 的相应处理部。应用程序脚本 32 将与特征信息相对应的广告信息的取得请求发送至应用程序提供者 30（步骤 1603）。

应用程序提供者 30 的广告分发功能 33，当通过 Web 服务器 31 接收到广告信息的取得请求时，从广告信息 34 中取得与接收到的特征信息相对应的信

息，返回作为请求源的 Web 浏览器 10（步骤 1604）。接收到广告信息的 Web 浏览器 10 将该信息传递至进行了广告信息的取得请求发送处理的应用程序脚本 32 的相应处理部。应用程序脚本 32 按照接收到的广告信息，在画面上显示广告信息（步骤 1605）。

例如，在步骤 1602 中，当作为内容的特征信息而由数据提供者 20 返回“web”时，在步骤 1603 中，应用程序脚本 32 将与“web”相对应的广告信息的取得请求发送至应用程序提供者 30。然后，在步骤 1604 中，作为与“web”相对应的广告信息，将“http://www.add.com/image/001.png”（参照图 17 的 1701）返回应用程序脚本 32，应用程序脚本 32 将“http://www.add.com/image/001.png”所表示的 URL 的图像显示在 Web 浏览器 10 上。

根据本实施例，可以提供与数据提供者 20 保存的数据紧密相关的、作为数据 API 功能需要不向外部公开的数据分析处理那样的功能的系统，可以提供与内容对应的广告分发的功能、或高速的检索处理的功能。

以往作为桌面应用程序而提供的应用程序，开始作为 Web 应用程序在因特网上公开。但是，从信息泄漏的观点考虑，企业用户较少使用这种 Web 应用程序。

对此，通过使用本发明，可以将 Web 应用程序的功能划分为与数据操作相关的功能、和与用户界面相关的功能的两部分，分别作为不同的域（不同的服务器）上的服务，因此，通过将提供与数据操作相关的功能的服务器配置在本公司内（或者可以信赖的公司外的组织），可以不直接向提供与用户界面相关的功能的服务器（提供该服务器的组织）传递数据，而将信息置于本公司内等，来利用应用程序的功能。

根据以上详细描述的本发明，若企业内用户的因特网上的 Web 应用程序利用规模扩大，则使用本发明的仅提供与用户界面相关的功能的服务、和提供或构建高可靠性的数据操作功能服务器的服务等事业会得到扩展。

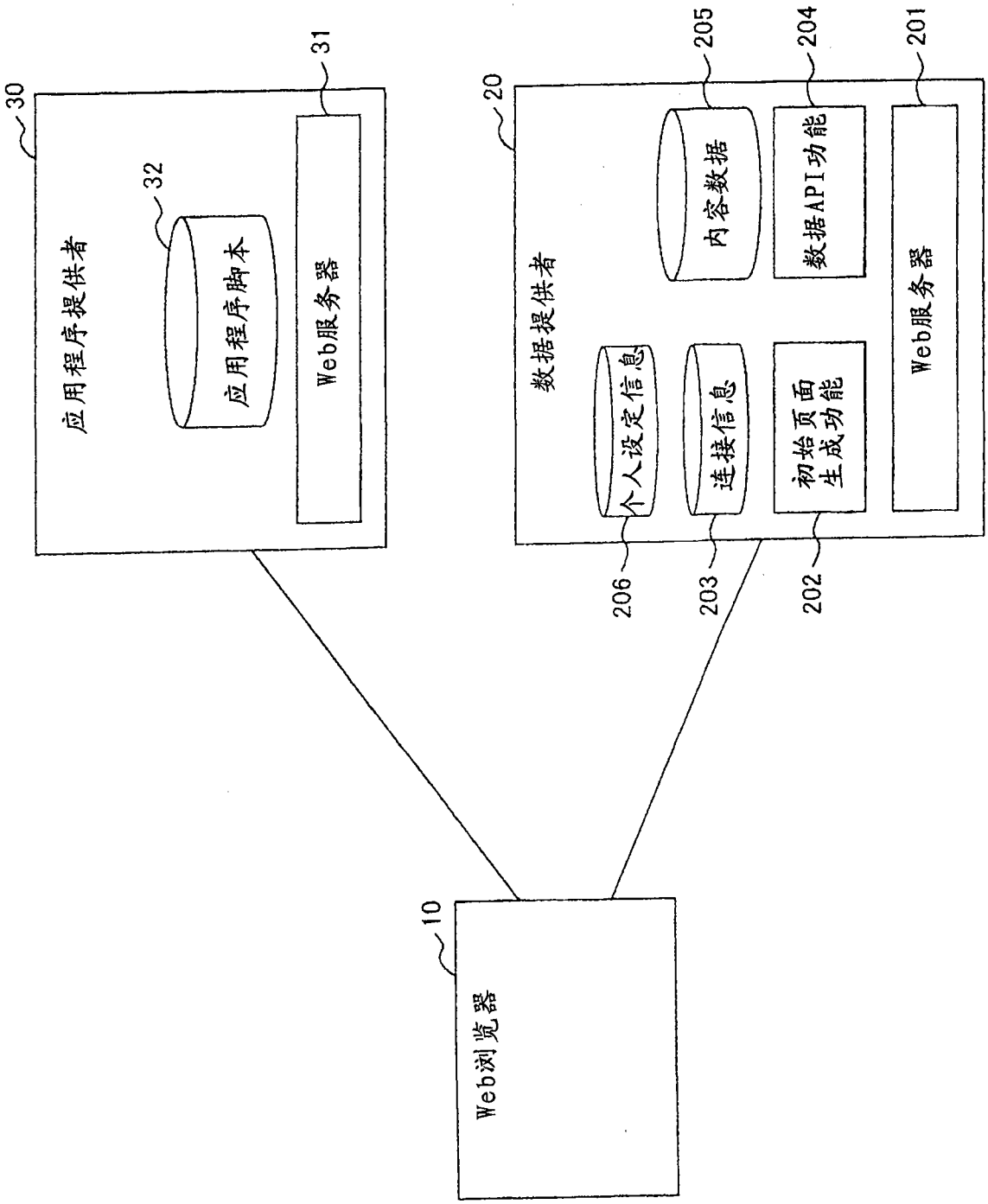

30
应用程序提供者
32
应用程序脚本
31
Web服务器
20
数据提供者
205
内容数据
204
数据API功能
201
Web服务器
个人设定信息
连接信息
初始页面生成功能
206
203
202
10
Web浏览器

图 1

```
1:  <html>
2:  <head>
3:  <title>カレンダーアプリ</title>
4:  <script src="http://appproviderA.com/script/calender.js"></script>
5:  <script>
6:  <!--//
7:  function init() {
8:      var urls={"get":"http://dataprovider.com/calendar/api/get/$user$",
9:                "add":"http://dataprovider.com/calendar/api/add/$user$",
10:               "edit":"http://dataprovider.com/calendar/api/adit/$user$"/$id$"};
11:     executeCalendar(urls,{"initView":"monthly","initDay":"Sunday","lang":"ja"});
12: }
13: //-->
14: </script>
15: </head>
16: <body onload="init();">
17: </body>
18: </html>
```

2001 2002 2003

图 2

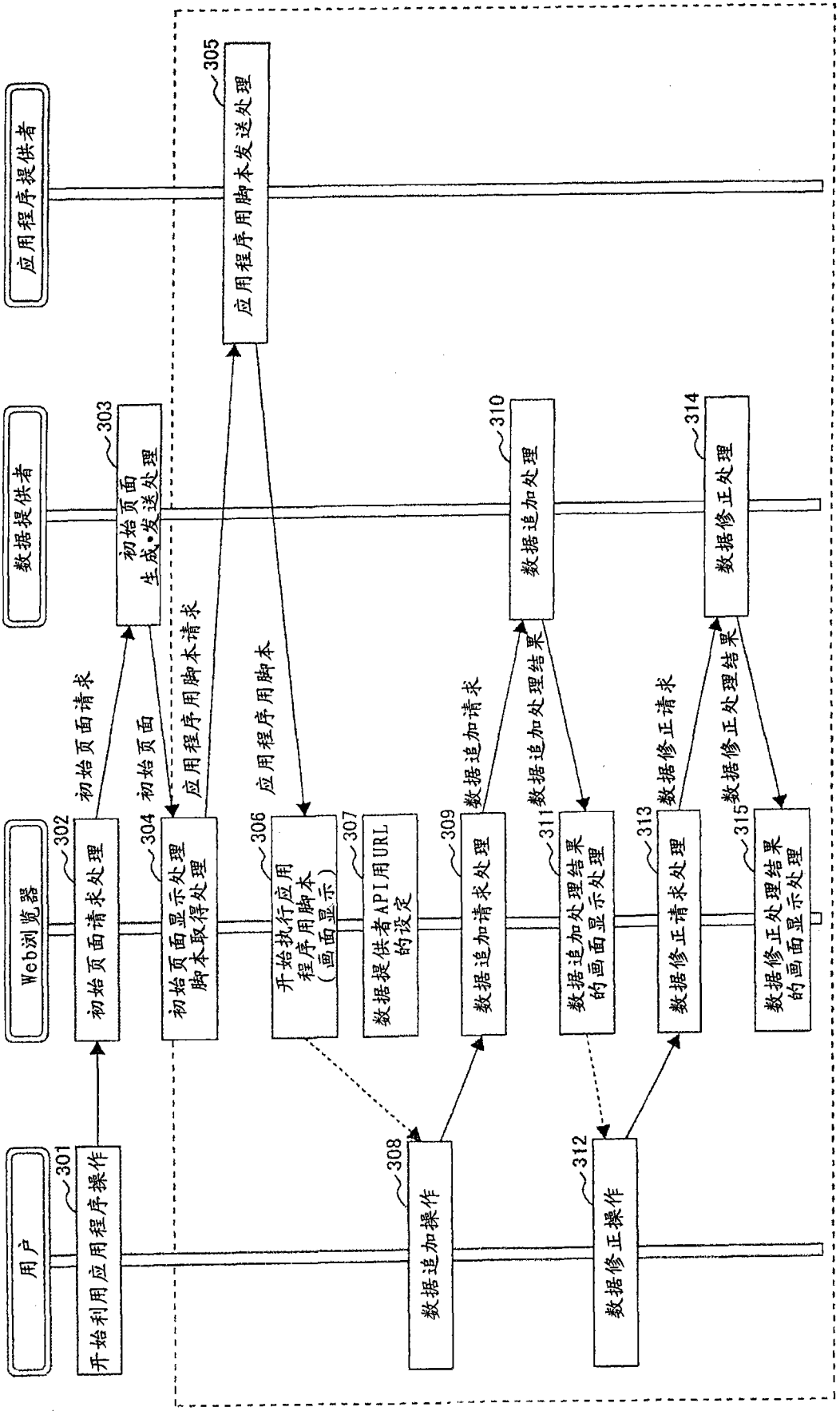

用户
Web浏览器
数据提供者
应用程序提供者
开始利用应用程序操作
301
初始页面请求
初始页面请求处理
302
初始页面
生成•发送处理
303
初始页面显示处理
脚本取得处理
304
应用程序用脚本请求
应用程序用脚本发送处理
305
应用程序用脚本
开始执行应用程序用脚本（画面显示）
306
数据提供者API用URL的设定
307
数据追加操作
308
数据追加请求处理
309
数据追加请求
数据追加处理
310
数据追加处理结果
数据追加处理结果的画面显示处理
311
数据修正操作
312
数据修正请求处理
313
数据修正请求
数据修正处理
314
数据修正处理结果
数据修正处理结果的画面显示处理
315

图 3

| API名 | 应用程序名 | 应用程序URL |
|---|---|---|
| calendar API | calendar app A | http://appproviderA.com/script/calendar.js |
| | Bcalendar | http://app.b.net/cal.js |
| blog API | blog app C | http://blogc.com/blog.js |
| | Dblog | http://www.d.net/app/dblog.js |

图 4

| API方法名 | API URL |
|---|---|
| GET | http://dataprovider.com/calendar/api/get/$user$ |
| ADD | http://dataprovider.com/calendar/api/add/$user$ |
| EDIT | http://dataprovider.com/calendar/api/edit/$user$/$id$ |
| ANALYZE | http://dataprovider.com/calendar/api/analyze/$user$ |

图 5

| 用户名 | 信息类别 | 设定值 | |
|---|---|---|---|
| Kojima | initialView | monthly | 601 |
| Kojima | initialDay | sunday | 602 |
| Kojima | language | japanese | 603 |

图 6

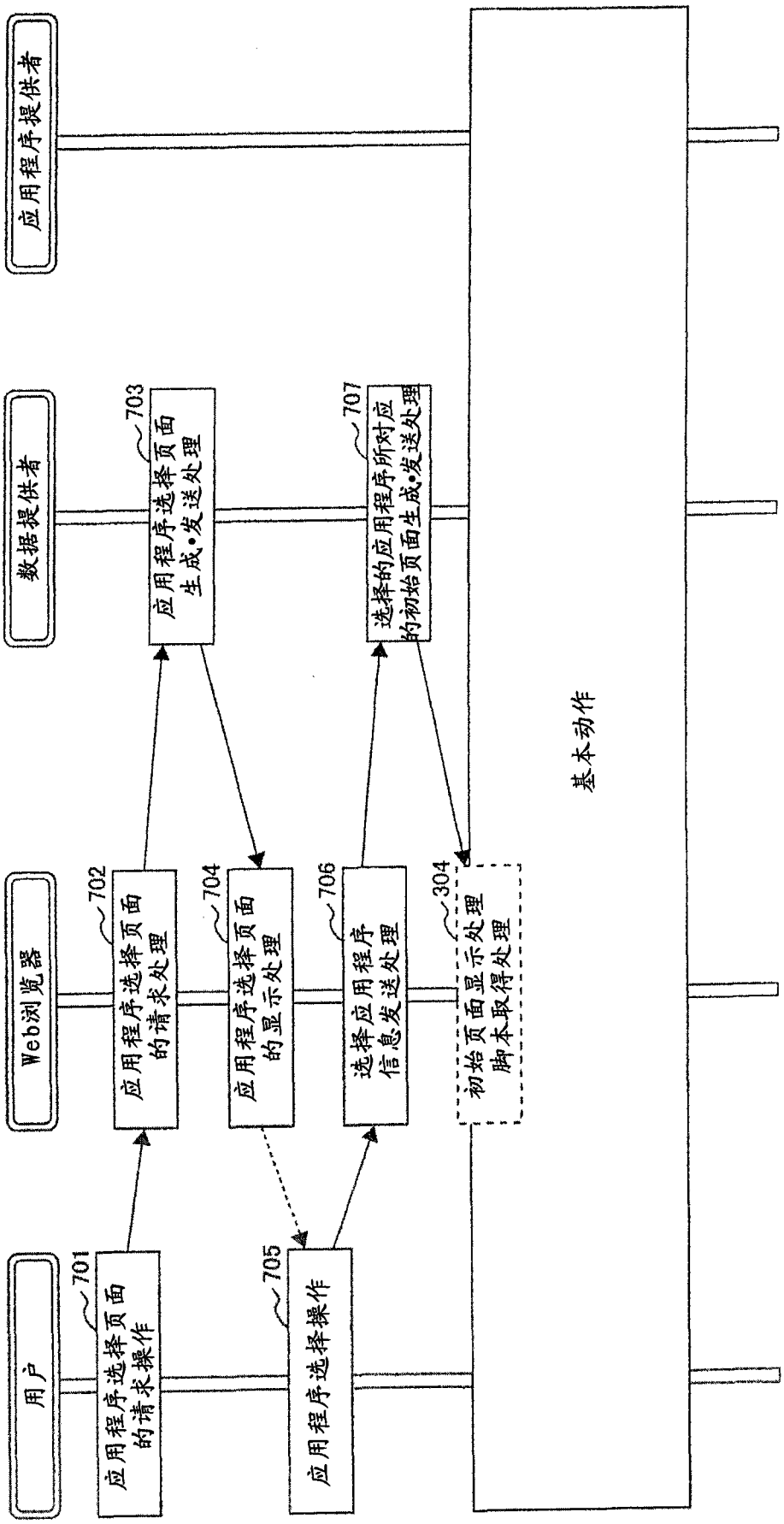

用户
Web浏览器
数据提供者
应用程序提供者
基本动作
701
应用程序选择页面的请求操作
702
应用程序选择页面的请求处理
703
应用程序选择页面生成•发送处理
704
应用程序选择页面的显示处理
705
应用程序选择操作
706
选择应用程序信息发送处理
707
选择的应用程序所对应的初始页面生成•发送处理
304
初始页面显示处理
脚本取得处理

图 7

| API名 | API方法名 | API URL |
| --- | --- | --- |
| calendar API | GET | http://dataprovider.com/calendar/api/get/$user$ |
| | ADD | http://dataprovider.com/calendar/api/add/$user$ |
| | EDIT | http://dataprovider.com/calendar/api/edit/$user$/$id$ |
| blog API | GET | http://dataprovider.com/blog/api/get/$user$ |
| | ADD | http://dataprovider.com/blog/api/add/$user$ |
| | EDIT | http://dataprovider.com/blog/api/edit/$user$/$id$ |

图 8

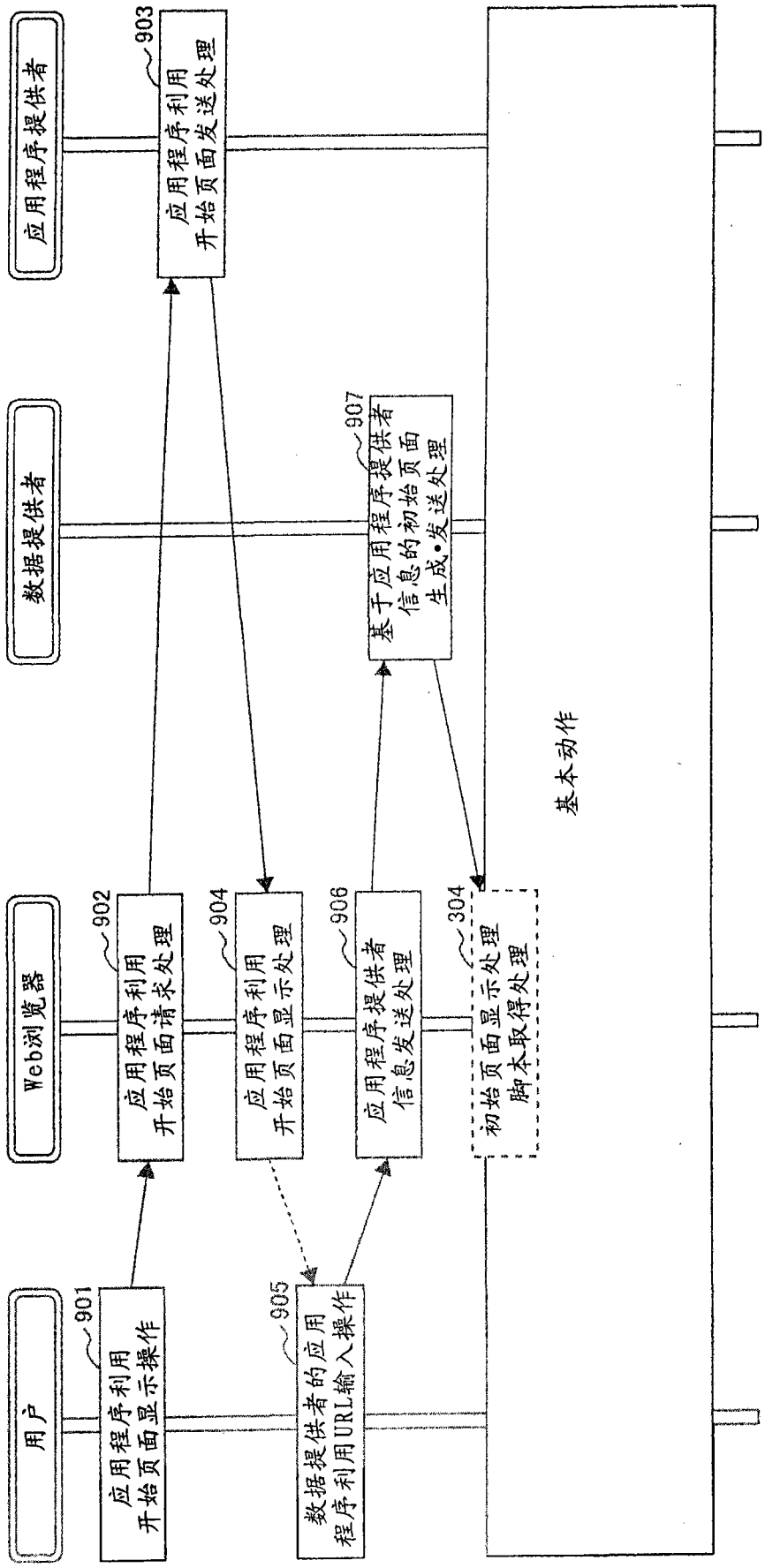
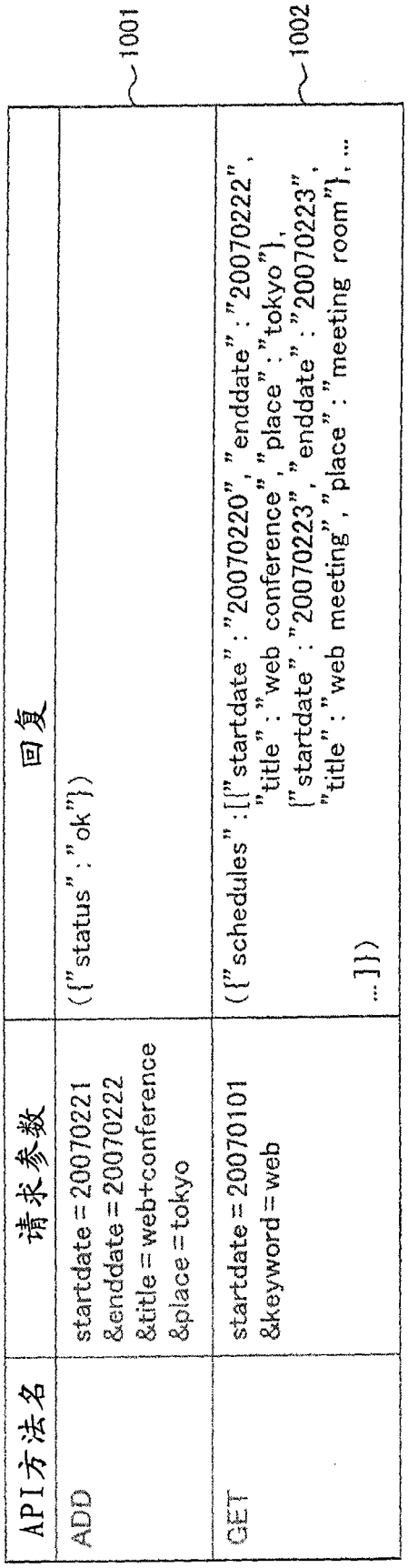

用户
Web浏览器
数据提供者
应用程序提供者
901
应用程序利用
开始页面显示操作
902
应用程序利用
开始页面请求处理
903
应用程序利用
开始页面发送处理
904
应用程序利用
开始页面显示处理
905
数据提供者的应用
程序利用URL输入操作
906
应用程序提供者
信息发送处理
907
基于应用程序提供者
信息的初始页面
生成•发送处理
304
初始页面显示处理
脚本取得处理
基本动作

图 9



| API方法名 | 请求参数 | 回复 |
| --- | --- | --- |
| ADD | startdate = 20070221<br>&enddate = 20070222<br>&title = web+conference<br>&place = tokyo | ({"status" : "ok"}) |
| GET | startdate = 20070101<br>&keyword = web | ({"schedules" :[{"startdate" : "20070220", "enddate" : "20070222",<br>"title" : "web conference", "place" : "tokyo"},<br>{"startdate" : "20070223", "enddate" : "20070223",<br>"title" : "web meeting", "place" : "meeting room"}, ...<br>...]}) |



图 10

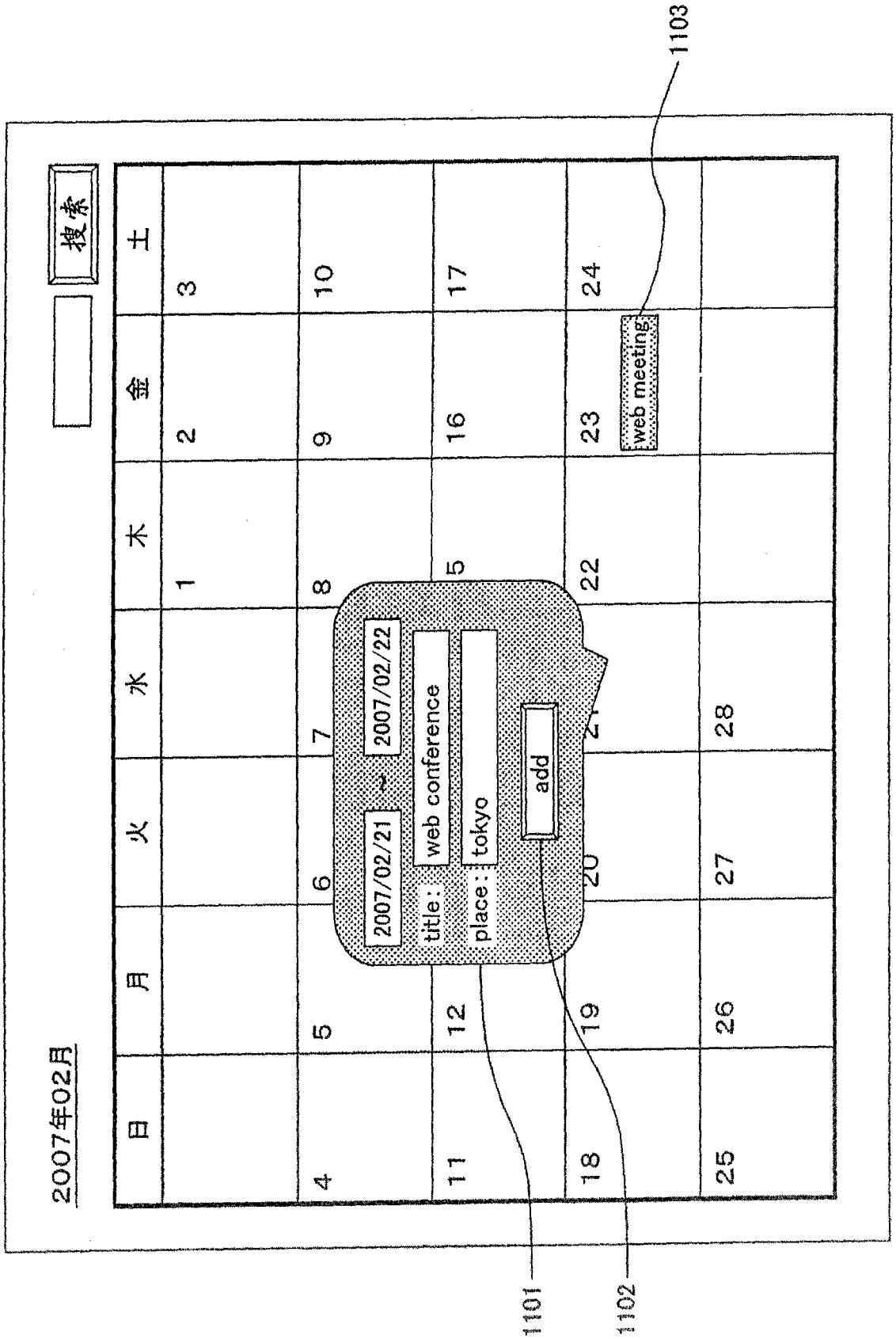

2007年02月
搜索
日 月 火 水 木 金 土
1 2 3
4 5 6 7 8 9 10
11 12 13 14 15 16 17
18 19 20 21 22 23 24
25 26 27 28
web meeting
2007/02/21
2007/02/22
title: web conference
place: tokyo
add
1101
1102
1103

图 11

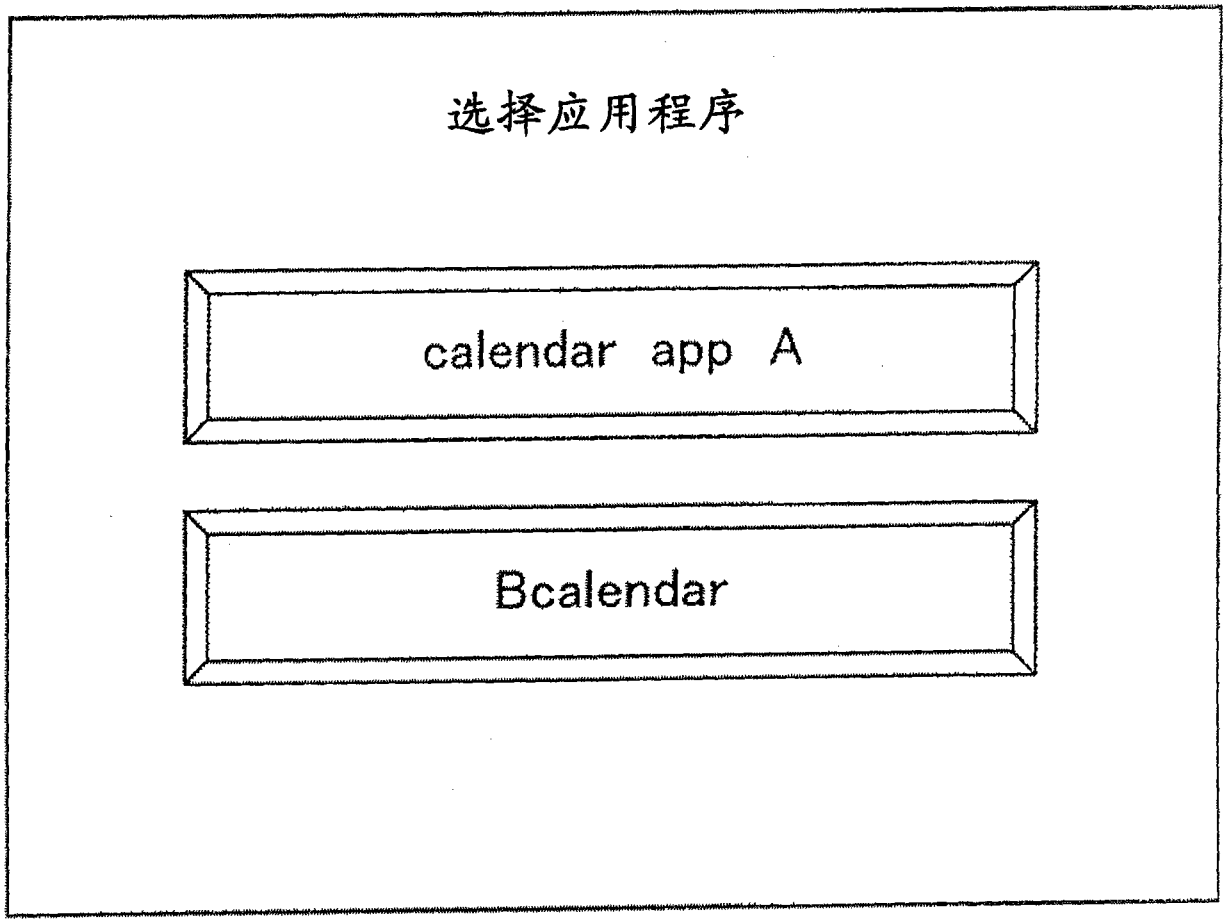



图 12

| 信息的种类 | URL |
| --- | --- |
| holiday | http://www.holidaycalendar.com/api/jsonp/” |
| weather | http://www.weatherinformation.com/jsonpapi/” |

1301

1302

图 13

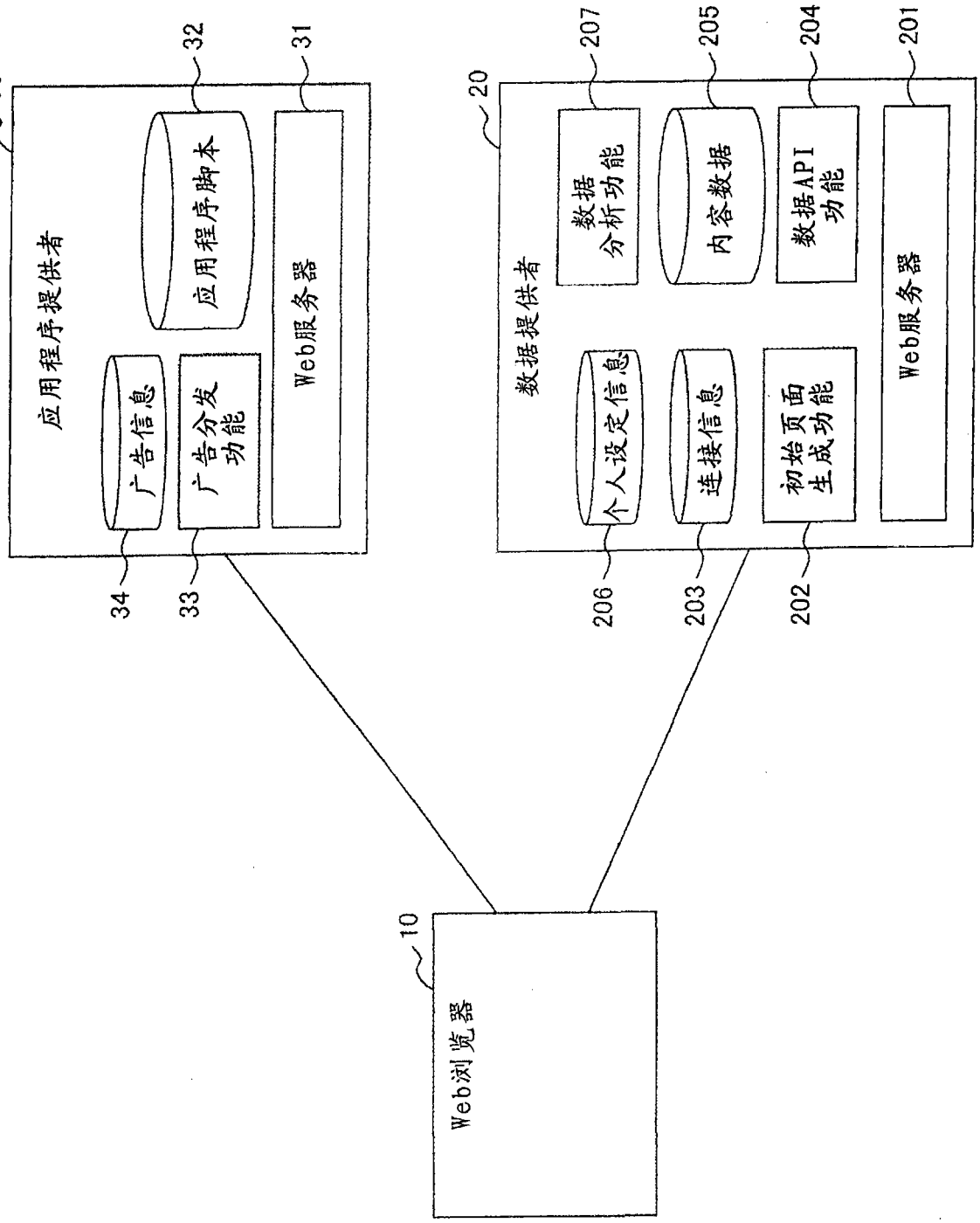

30
应用程序提供者
32
应用程序脚本
31
Web服务器
广告信息
广告分发功能
34
33
20
数据提供者
207
数据分析功能
205
内容数据
204
数据API功能
201
Web服务器
个人设定信息
连接信息
初始页面生成功能
206
203
202
10
Web浏览器

图 14

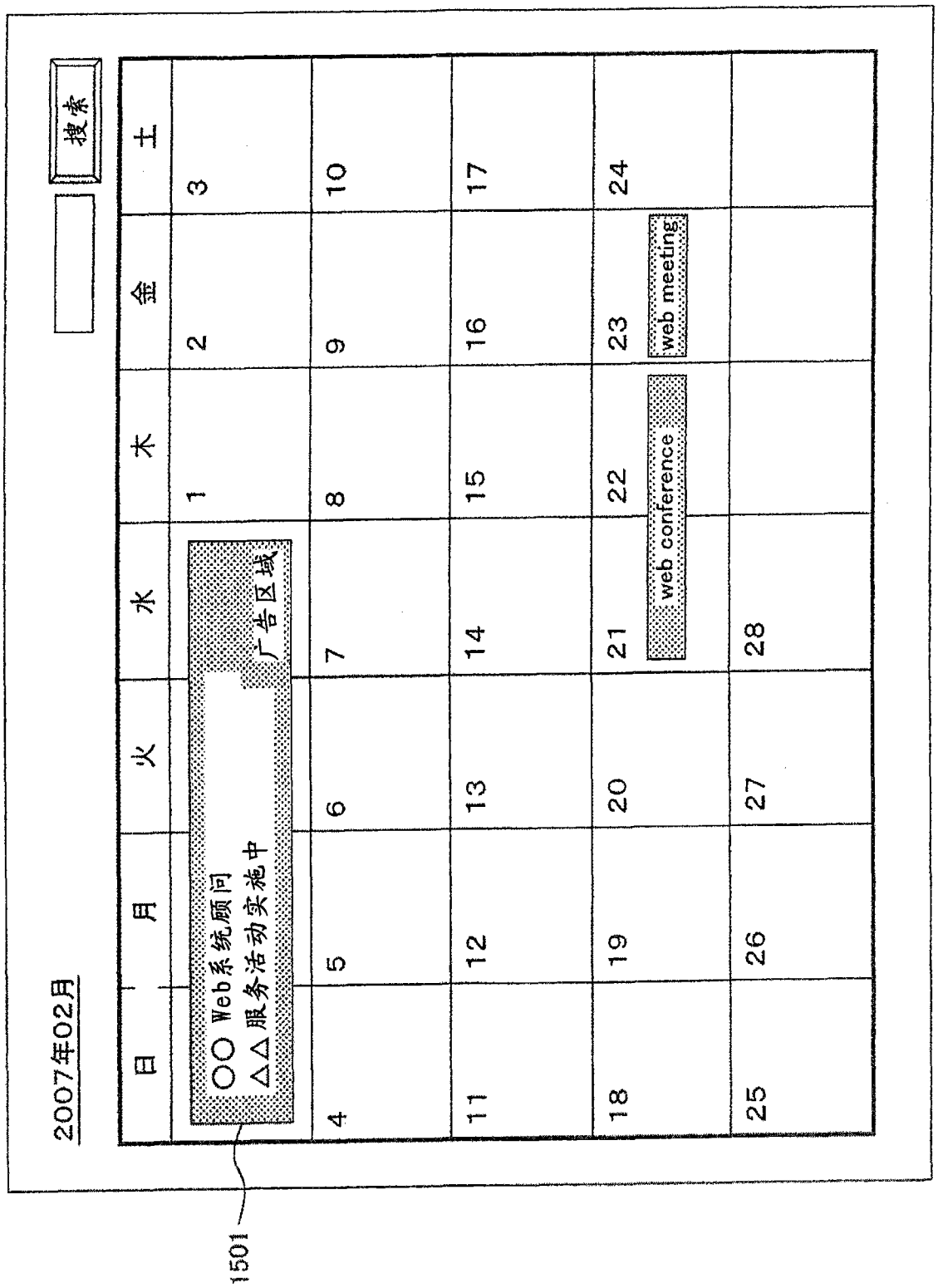

2007年02月
搜索
日 月 火 水 木 金 土
1 2 3
4 5 6 7 8 9 10
11 12 13 14 15 16 17
18 19 20 21 22 23 24
25 26 27 28
○○ Web系统顾问
△△服务活动实施中
广告区域
web conference
web meeting
1501

图 15

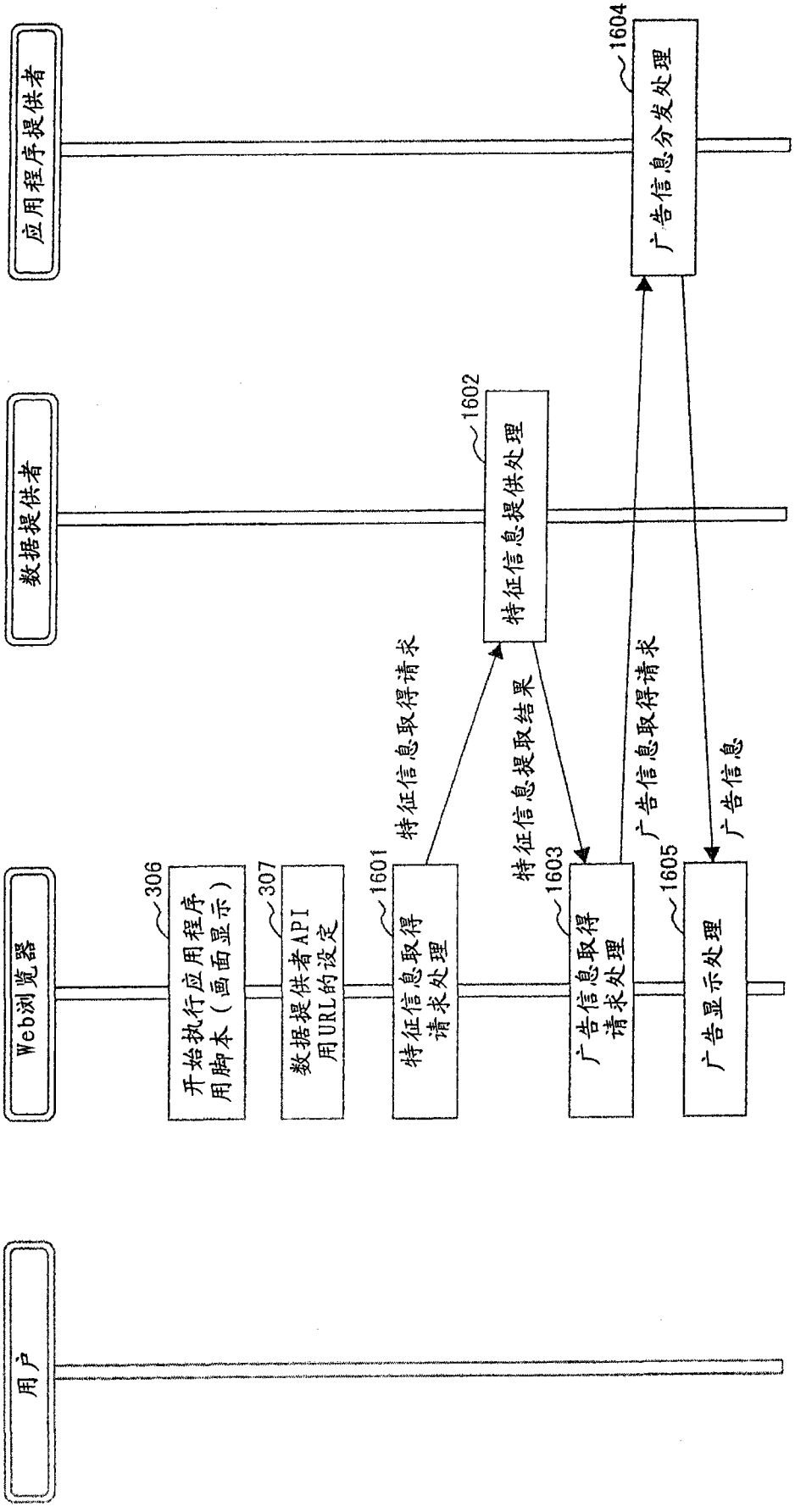

用户
Web浏览器
数据提供者
应用程序提供者
306
开始执行应用程序用脚本（画面显示）
307
数据提供者API用URL的设定
1601
特征信息取得请求处理
特征信息取得请求
1602
特征信息提供处理
特征信息提取结果
1603
广告信息取得请求处理
广告信息取得请求
1604
广告信息分发处理
1605
广告显示处理
广告信息

图 16

| 特征信息 | 广告URL | |
|---|---|---|
| web | http://www.add.com/image/001.png | 1701 |
| network | http://www.add.com/image/002.png | 1702 |
| application | http://www.add.com/image/003.png | 1703 |

图 17

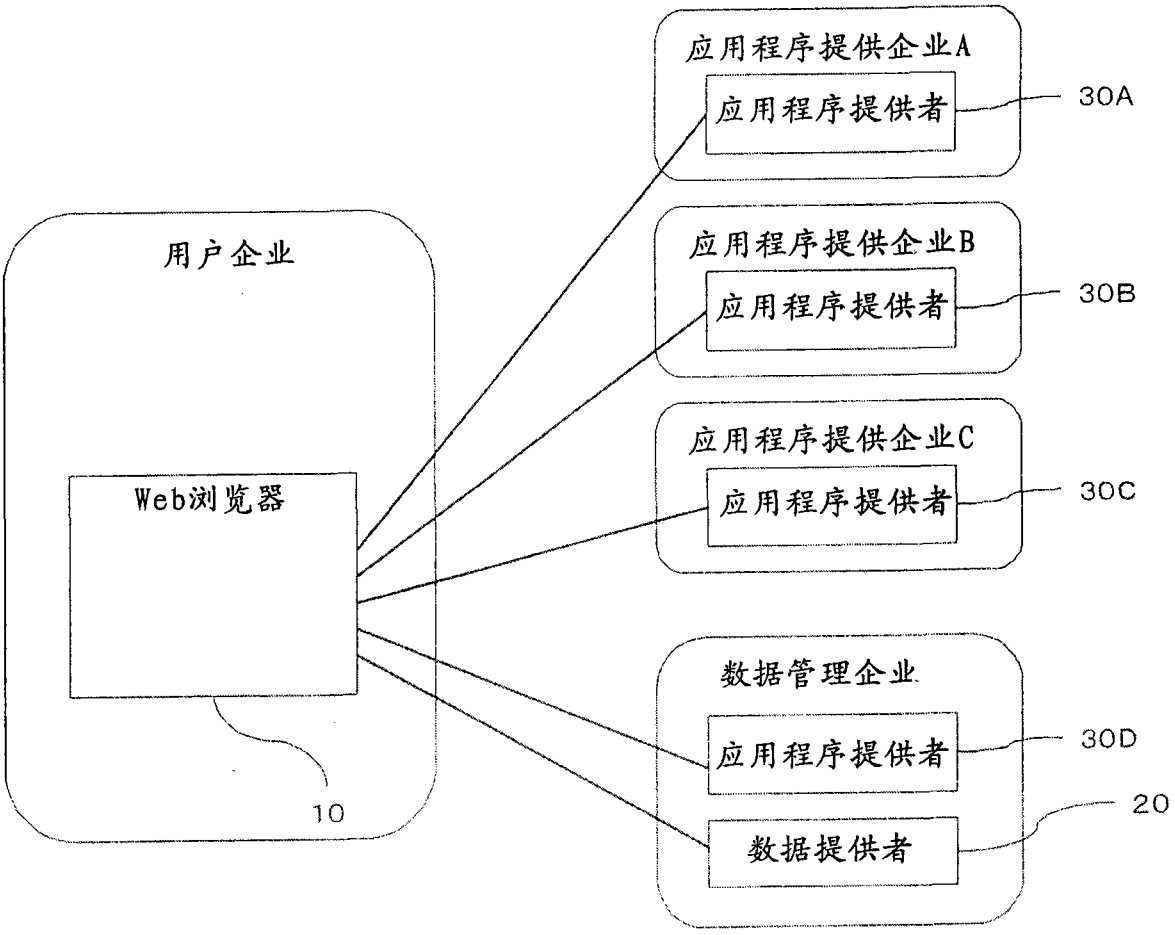



图 18