



(12) 发明专利申请

(10) 申请公布号 CN 104066623 A

(43) 申请公布日 2014. 09. 24

(21) 申请号 201280068034. 3

(51) Int. Cl.

(22) 申请日 2012. 01. 25

B60R 16/02(2006. 01)

(85) PCT国际申请进入国家阶段日  
2014. 07. 24

(86) PCT国际申请的申请数据  
PCT/JP2012/000459 2012. 01. 25

(87) PCT国际申请的公布数据  
W02013/111185 JA 2013. 08. 01

(71) 申请人 三菱电机株式会社  
地址 日本东京

(72) 发明人 水口武尚 渡边义明 泷本康明  
三井武史 中井良弘

(74) 专利代理机构 上海专利商标事务所有限公  
司 31100

代理人 陈力奕

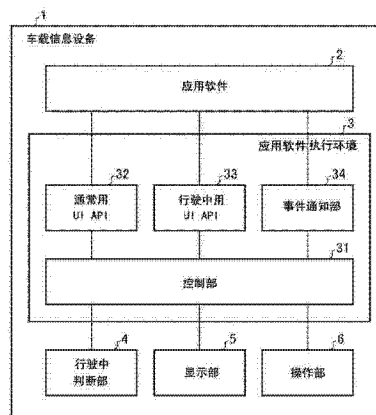
权利要求书2页 说明书17页 附图16页

(54) 发明名称

移动体用信息设备

(57) 摘要

本发明包括：通常用 UI API (32)，该通常用 UI API (32) 生成由应用软件 (2) 指定的画面结构的画面数据；行驶中用 UI API (33)，该行驶中用 UI API (33) 基于规定车辆行驶过程中所显示的行驶中用的画面结构的布局的模板数据，生成由应用软件 (2) 指定的车辆行驶过程中所显示的行驶中用画面结构的画面数据；以及控制部 (31)，该控制部 (31) 设置于应用软件执行环境 (3)，在车辆处于停止时，在显示部 (5) 中显示由通常用 UI API (32) 生成的画面数据，在车辆正在行驶时，在显示部 (5) 中显示由行驶中用 UI API (33) 生成的画面数据。



1. 一种移动体用信息设备,具备进行画面显示的显示部、以及执行应用软件的应用软件执行环境,其特征在于,包括:

第一 API(Application Program Interface:应用软件程序接口),该第一 API 生成由所述应用软件指定的画面结构的画面数据;

第二 API,该第二 API 基于对移动体在移动中所显示的移动中用画面结构的布局进行了规定的模板数据,来生成由所述应用软件指定的所述移动中用画面结构的画面数据;以及

控制部,该控制部设置于所述应用软件执行环境,在所述移动体处于停止时,在所述显示部中显示由所述第一 API 生成的所述画面数据,在所述移动体正在移动时,在所述显示部中显示由所述第二 API 生成的所述画面数据。

2. 如权利要求 1 所述的移动体用信息设备,其特征在于,

所述应用软件执行环境具有多个模板数据,该多个模板数据分别规定所述移动体在移动中所显示的多个画面结构的多个布局,

所述第二 API 基于根据所述应用软件的指定内容而从所述多个模板数据中选择出的模板数据,来生成所述移动体在移动中所显示的多个画面结构的画面数据。

3. 如权利要求 1 所述的移动体用信息设备,其特征在于,

所述第二 API 根据所述应用软件的指令,对构成所述模板数据所规定的画面结构的布局的显示要素进行变更,从而生成所述移动中用的画面结构的画面数据。

4. 如权利要求 1 所述的移动体用信息设备,其特征在于,

所述第二 API 根据所述应用软件的指令,在规定的限制范围内对构成所述移动中用的画面的显示要素的形态进行变更,其中,所述移动中用的画面基于所述模板数据而生成。

5. 如权利要求 1 所述的移动体用信息设备,其特征在于,

所述第一 API 在所述移动体处于停止时生成所述画面数据,

所述第二 API 在所述移动体正在移动时生成所述移动中用的画面结构的画面数据。

6. 如权利要求 1 所述的移动体用信息设备,其特征在于,

包括离屏缓存,该离屏缓存保存对所述画面数据进行绘制处理而得到的绘制数据,

所述控制部以使得由所述第一 API 生成的画面数据的绘制数据、和由所述第二 API 生成的画面数据的绘制数据位于不同显示层的方式,来将它们存储在所述离屏缓存中,并根据所述移动体是否正在移动,来切换保存在所述离屏缓存中的所述各绘制数据,并在所述显示部进行显示。

7. 如权利要求 1 所述的移动体用信息设备,其特征在于,

包括语音操作部,该语音操作部对用户发出的语音进行识别,在识别结果与由所述控制部登记的语音命令相一致或相类似的情况下,将该识别结果作为语音事件通知给所述控制部,

所述第二 API 生成编入有所述语音命令的移动中用的画面结构的画面数据。

8. 一种移动体用信息设备,具备进行画面显示的显示部、以及执行应用软件的应用软件执行环境,其特征在于,包括:

第一 API(Application Program Interface:应用软件程序接口),该第一 API 生成由所述应用软件指定的画面结构的画面数据;

移动中用 UI 生成部,该移动中用 UI 生成部基于由所述第一 API 生成的画面数据,生成由所述应用软件指定的、在所述移动体移动中进行显示的移动中用的画面结构的画面数据;以及

控制部,该控制部设置于所述应用软件执行环境,在所述移动体处于停止时,在所述显示部中显示由所述第一 API 生成的画面数据,在所述移动体正在移动时,在所述显示部中显示由所述移动中用 UI 生成部生成的画面数据。

9. 如权利要求 8 所述的移动体用信息设备,其特征在于,

所述移动中用 UI 生成部在由所述第一 API 生成的画面数据的画面中包含有视频的情况下,生成将该视频转换成静止图像后得到的画面结构的画面数据。

10. 如权利要求 8 所述的移动体用信息设备,其特征在于,

所述第一 API 生成作为附带信息而包含有以下信息的画面数据,所述信息构成所述移动中用的画面结构的画面数据,

所述移动中用 UI 生成部基于由所述第一 API 生成的画面数据中的所述附带信息,来生成所述移动中用的画面结构的画面数据。

11. 如权利要求 8 所述的移动体用信息设备,其特征在于,

包括离屏缓存,该离屏缓存保存对所述画面数据进行绘制处理而得到的绘制数据,

所述控制部以使得由所述第一 API 生成的画面数据的绘制数据、和由所述移动中用 UI 生成部生成的画面数据的绘制数据位于不同显示层的方式,来将它们存储在所述离屏缓存中,并根据所述移动体是否正在移动,来切换保存在所述离屏缓存中的所述各绘制数据,并在所述显示部进行显示。

12. 如权利要求 8 所述的移动体用信息设备,其特征在于,

包括语音操作部,该语音操作部对用户发出的语音进行识别,在识别结果与由所述控制部登记的语音命令相一致或相类似的情况下,将该识别结果作为语音事件通知给所述控制部,

所述移动中用 UI 生成部生成编入有所述语音命令的移动中用的画面结构的画面数据。

13. 如权利要求 1 或 8 所述的移动体用信息设备,其特征在于,

包括多个显示部,

所述控制部在所述移动体处于停止时,在所述多个显示部中的规定的显示部中显示由所述第一 API 生成的所述画面数据,在所述移动体正在移动时,在所述规定的显示部中显示所述移动中用的画面结构的画面数据,

无论所述移动体是否正在移动,所述控制部均在所述多个显示部中的所述规定的显示部以外的显示部中显示由所述第一 API 生成的所述画面数据。

## 移动体用信息设备

### 技术领域

[0001] 本发明涉及搭载于车辆等移动体、并具备显示应用软件图像的显示部的移动体用信息设备。

### 背景技术

[0002] 在搭载于车辆等的信息设备中,为了不妨碍驾驶员所进行的车辆的驾驶,需要对车辆在行驶过程中的画面显示以及基于该画面显示的操作的内容进行限制。例如,在非专利文献 1 中记载有:应该要对车辆用信息设备进行优化,以使得驾驶员能在短时间内确认画面上显示的信息量。

[0003] 此外,在专利文献 1 中揭示了以下车载装置,该车载装置具备基于画面显示进行输入操作的触摸面板等接触式输入单元、以及通过移动拨盘开关等画面上的聚焦点来进行选择操作的移动式输入单元。

该装置中,在车辆处于停止时,在显示装置上显示由适合通过触摸面板输入的菜单项目排列而构成的菜单画面,在车辆正在行驶时,在显示装置上显示由适合通过拨盘开关输入的菜单项目排列而构成的菜单画面。

由此,在专利文献 1 中,通过预先准备适合车辆处于停止的情况的菜单画面和适合车辆行驶中的情况的菜单画面,根据车辆的状态来切换菜单画面,从而提高了选择菜单项目的操作性。

[0004] 另一方面,随着近年来车载信息设备的通信功能和信息处理能力趋于高性能化,希望将由车载信息设备的制造厂商以外的第三方开发的应用软件(以下记载为第三方应用软件)下载到车载信息设备并加以利用的需求正在增加。

在这种情况下,车载信息设备的制造厂商需要使第三方应用软件也遵守车辆在行驶时的操作内容限制。

现有技术文献

专利文献

[0005] 专利文献 1:日本专利特开 2008-65519 号公报

非专利文献

[0006] 非专利文献 1:“图像显示装置指南关于图像显示装置的使用修订第 3.0 版”、日本汽车工业协会、平成 16 年 8 月 18 日

### 发明内容

发明所要解决的技术问题

[0007] 第三方应用软件的画面显示、操作的接受等 UI(User Interface:用户界面)是使用车载信息设备所提供的 API(Application Program Interface:应用软件程序接口)来进行开发的。API 中能够指定构成字符串、图像、按钮等画面的显示要素,通常情况下,能够自由地配置显示要素,还能够指定尺寸。因此,在第三方应用软件不是设计用于车载的情

况下,可以不考虑车辆处于停止中还是正在行驶中的情况,自由地在画面上显示字符串、图像、按钮等。

[0008] 另一方面,对于确认第三方应用软件是否遵守车辆在行驶时的操作内容的限制,需要对第三方应用软件在车载信息设备上的所有动作进行试验并确认。因此,车载信息设备的制造厂商要对所有的第三方应用软件实施上述确认是非常困难的。

[0009] 因此,若车辆处于行驶过程中则禁止第三方应用软件的操作,这样的话车载信息设备的制造厂商就能省略确认操作。

然而,有时即使在车辆处于行驶过程中时,也希望能够以不妨碍行驶为前提来浏览较少的信息,或者进行简单操作,因而,若在车辆处于行驶过程中时一律禁止操作,则显著损害了用户的便利性。

[0010] 此外,在专利文献 1 所代表的现有技术中,由于以预先准备适合车辆处于停止的情况的菜单画面和车辆行驶中的情况的菜单画面为前提,因此直接应用由车载信息设备的制造厂商以外的第三方开发的第三方应用软件是不可能的。并且,专利文献 1 中,以在制造车载装置时预先安装的应用软件为前提,因此,对于将第三方应用软件中的画面显示、操作内容切换为适合车辆行驶中的情况的设想是不现实的。

[0011] 本发明是为了解决上述问题而完成的,其目的在于获得一种能够显示适合移动体移动过程中的画面的移动体用信息设备。

解决技术问题所采用的技术方案

[0012] 本发明所涉及的移动体用信息设备包括:第一 API,该第一 API 生成由应用软件指定的画面结构的画面数据;第二 API,该第二 API 基于对移动体在移动中所显示的移动中用画面结构的布局进行了规定的模板数据,生成由应用软件指定的移动中用画面结构的画面数据;以及控制部,该控制部设置在应用软件执行环境中,在移动体处于停止时,在显示部中显示由第一 API 生成的画面数据,在移动体正在移动时,在显示部中显示由第二 API 生成的画面数据。

发明效果

[0013] 根据本发明,具有能够显示适合移动体移动过程中的画面的效果。

## 附图说明

[0014] 图 1 是表示本发明的实施方式 1 所涉及的移动体用信息设备的结构的框图。

图 2 是表示以 HTML(Hyper Text Markup Language:超文本链接标记语言)格式呈现车辆处于停止时的画面结构的画面数据的一个示例的图。

图 3 是表示基于图 2 的画面数据进行显示的画面的图。

图 4 是表示以 XML(eXtensible Markup Language:可扩展标记语言)格式呈现车辆正在行驶时的画面结构的画面数据的一个示例的图。

图 5 是表示基于图 4 的画面数据进行显示的画面的图。

图 6 是表示以 XML 格式呈现车辆正在行驶时的画面结构的画面数据的另一个示例的图。

图 7 是表示基于图 6 的画面数据进行显示的画面的图。

图 8 是表示实施方式 1 所涉及的移动体用信息设备的动作的流程图。

图 9 是表示本发明的实施方式 2 所涉及的移动体用信息设备的动作的流程图。

图 10 是表示本发明的实施方式 3 所涉及的移动体用信息设备的动作的流程图。

图 11 是表示实施方式 3 中车辆正在行驶时的显示画面的一个示例的图。

图 12 是表示本发明的实施方式 4 所涉及的移动体用信息设备的结构的框图。

图 13 是表示实施方式 4 所涉及的移动体用信息设备的动作的流程图。

图 14 是表示以 XML 格式呈现车辆正在行驶时的画面结构的画面数据的一个示例的图。

图 15 是表示以 HTML 格式呈现车辆处于停止时的画面结构的画面数据的一个示例的图。

图 16 是表示基于图 15 的画面数据进行显示的画面的图。

图 17 是表示以 XML 格式呈现车辆正在行驶时的画面结构的画面数据的另一个示例的图。

图 18 是表示基于图 17 的画面数据进行显示的画面的图。

图 19 是表示以 HTML 格式呈现车辆处于停止时的画面结构的画面数据的另一个示例的图。

图 20 是表示本发明的实施方式 5 所涉及的移动体用信息设备的结构的框图。

图 21 是表示实施方式 5 所涉及的移动体用信息设备的动作的流程图。

图 22 是表示以 XML 格式呈现车辆正在行驶时的画面结构的画面数据的一个示例的图。

## 具体实施方式

[0015] 下面,为了详细说明本发明,根据附图,对用于实施本发明的方式进行说明。

### 实施方式 1.

图 1 是表示本发明的实施方式 1 所涉及的移动体用信息设备的结构的框图,示出了将实施方式 1 所涉及的移动体用信息设备应用于车载信息设备的情况。图 1 所示的车载信息设备 1 中设置有执行应用软件 2 的应用软件执行环境 3、行驶中判断部 4、显示部 5 以及操作部 6。

应用软件 2 是由应用软件执行环境 3 来使其动作的软件,是执行与各种目的用途相对应的处理的软件,例如进行车载信息设备 1 的监视 / 控制的软件、进行导航处理的软件、进行游戏的软件等。

另外,应用软件 2 的程序可以预先存储在车载信息设备 1 的内部(图 1 中未图示的存储装置),也可以通过网络从外部下载,还可以从 USB(Universal Serial Bus:通用串行总线)存储器等外部存储介质进行安装。

[0016] 应用软件执行环境 3 是使应用软件 2 进行动作的执行环境,从其功能上来看包括控制部 31、通常用 UI API32、行驶中用 UI API33、以及事件通知部 34。

控制部 31 是对用于使应用软件 2 动作的整体动作进行控制的控制部。此外,控制部 31 具有根据搭载有车载信息设备 1 的车辆处于停止时所显示的画面结构(以下称为通常用画面结构)的画面数据来绘制通常用画面的功能、以及基于该车辆行驶中所显示的画面结构(以下称为行驶中用画面结构)的画面数据来绘制行驶中用画面的功能。

[0017] 通常用 UI API32 是用于根据应用软件 2 来指定通常用画面结构的 API。在利用应用软件 2 的处理进行画面显示时将该通常用 UI API32 提供给应用软件 2,生成由应用软件

2 指定的通常用画面结构的画面数据。

行驶中用 UI API33 是用于根据应用软件 2 来指定行驶中用画面结构的 API。在利用应用软件 2 的处理进行画面显示时将该行驶中用 UI API33 提供给应用软件 2,生成由应用软件 2 指定的行驶中用画面结构的画面数据。另外,行驶中用 UI API33 与通常用 UI API32 相比,其画面结构的指定存在限制,设为仅可指定适合车辆行驶中的画面结构。

此外,事件通知部 34 将车辆的行驶状态的变化、利用操作部 6 的用户操作事件等事件通知给应用软件 2。

[0018] 行驶中判断部 4 与搭载于车辆的车速传感器等相连接,判断该车辆是正在行驶中还是处于停止中,并将判断结果作为行驶状态变化事件通知给应用软件执行环境 3。

显示部 5 是进行画面显示的显示装置,例如是液晶显示器等显示装置。显示部 5 在画面上显示通过控制部 31 的绘制处理而得到的画面的绘制数据。

操作部 6 是接受来自用户的操作的操作部,例如通过设置在显示部 5 的画面上的触摸面板或硬件按键、显示在画面上的软件按键等来实现。

[0019] 图 2 是表示以 HTML 格式呈现车辆处于停止时的画面结构(通常用画面结构)的画面数据的一个示例的图,使用通常用 UI API32 来指定。此外,图 3 是表示基于图 2 的画面数据进行显示的画面的图。

图 2 所示的示例中,描述有 5 个在画面内绘制矩形的 <div> 要素、以及 4 个 <button> 要素。此外,通过这些要素的 <style> 要素内以 CSS(Cascading Style Sheet:层叠样式表)格式进行描述的 padding,margin,border,width,height,background 等类型指定,来指定各要素的类型。

应用软件 2 根据操作事件的内容,确定构成通常用画面的显示要素(字符串、图像、按钮等)的配置、尺寸、字体、字体大小、字符数等,在通常用 UI API32 中指定图 2 所示那样的通常用画面结构。通常用 UI API32 按照应用软件 2 的指定内容,生成以用于在应用软件执行环境 3 内进行处理的内部数据格式来呈现通常用画面结构的画面数据。该内部数据格式用于保持画面数据,以便于应用软件执行环境 3 进行处理,其格式是任意的。作为该内部数据格式的示例,例如为作为用于利用计算机程序处理 HTML、XML 的格式而已知的 DOM(Document Object Model:文件对象模型、http://www.w3.org/DOM/)。由于 DOM 只是将 HTML、XML 转换成计算机程序易于处理的数据格式而得到的,因此在以下的画面数据的说明中,以 HTML 或 XML 格式来进行说明。

该画面数据从通常用 UI API32 传输至应用软件执行环境 3 的控制部 31。控制部 31 对从通常用 UI API32 接收到的画面数据进行分析,并基于该分析结果按照绘制命令来进行通常用画面的绘制处理。显示部 5 输入由控制部 31 生成的绘制数据,并显示图 3 所示的画面。

[0020] 图 4 是表示以 XML 格式呈现车辆正在行驶时的画面结构(行驶中用画面结构)的画面数据的一个示例的图,使用行驶中 UI API33 来指定。此外,图 5 是表示基于图 4 的画面数据进行显示的画面的图。

图 4 所示的示例是与图 3 所示的通常用画面相对应的行驶中用画面的画面数据,并且示出进行基于“template-A”的内容的画面显示。

这里,“template-A”是预先在行驶中 UI API33 中准备的画面结构,显示页标头(图 5 中显示为“新闻:标题”)、“行驶中无法显示”的消息字符串和两个按钮。

此外,在该图 4 的示例中,行驶中用 UI API33 根据应用软件 2 的指令,通过 <text> 要素,将由“msg1”规定的页标头的字符串替换为“新闻:标题”,将由“btn2”规定的按钮的字符串替换为“语音朗读”。

[0021] 另外,在应用软件执行环境 3 中,预先准备有规定行驶中用画面的布局的模板数据。

应用软件 2 根据操作事件的内容确定构成行驶中用画面的显示要素,并指定给行驶中用 UI API33。行驶中用 UI API33 选择上述行驶中用画面的模板数据 (“template-A”),基于由应用软件 2 指定的显示要素,从图 4 所示那样的行驶中用画面结构来生成画面数据。该画面数据从行驶中用 UI API33 传输至应用软件执行环境 3 的控制部 31。

控制部 31 对从行驶中用 UI API33 接收到的画面数据进行分析,并根据基于该分析结果的绘制命令来进行行驶中用画面的绘制处理。显示部 5 输入由控制部 31 生成的绘制数据,并显示图 5 所示的画面。

[0022] 图 5 中,例如从图 3 的通常用画面的显示要素中省略了“ABC 获胜!”、“日元进一步走高”、“DEF 公司和 GHI 公司合作”,并且还省略了“前一页”和“后一页”的按钮。

但是,在本发明中,不会像现有技术那样在车辆行驶中一律不允许进行画面操作,而在利用一次操作可完成处理这样的分散驾驶员的注意力的可能性较低的操作的情况下,保留显示与其画面操作相应的显示要素。例如,在图 5 中显示用于使画面转移至前一画面的“返回”按钮、仅用于利用语音来进行信息的朗读的“语音朗读”的按钮。

[0023] 图 6 是表示以 XML 格式呈现车辆正在行驶时的画面结构 (行驶中用画面结构) 的画面数据的另一个示例的图,使用行驶中用 UI API33 来指定。此外,图 7 是表示基于图 6 的画面数据进行显示的画面的图。

图 6 所示的示例是与图 3 的通常用画面相对应的呈现行驶中用画面的画面数据,并且示出进行基于“template-B”的画面显示的情况。

这里,“template-B”是预先在行驶中 UI API33 中准备的画面结构,在画面中显示由“msg1”这样的识别符来表示的字符串、以及“是”和“否”这样的按钮。

此外,在该图 6 所示的示例中,行驶中用 UI API33 根据应用软件 2 的指令,通过 <text> 要素,将由“msg1”规定的页标头字符串替换为“执行 abc?”这样的字符串。

[0024] 这里,行驶中用 UI API33 选择行驶中用画面的模板数据 (“template-B”),基于由应用软件\_2 指定的显示要素,由以图 6 所示那样的 XML 格式呈现的行驶中用画面结构来生成画面数据。该画面数据从行驶中用 UI API33 传输至应用软件执行环境 3 的控制部 31。控制部 31 对从行驶中用 UI API33 接收到的画面数据进行分析,并根据基于该分析结果的绘制命令来进行行驶中用画面的绘制处理。显示部 5 输入由控制部 31 生成的绘制数据,并显示图 7 所示的画面。

[0025] 如上所述,为了构成图 4 和图 6 那样的画面数据,行驶中用 UI API33 准备规定适合车辆行驶中的画面结构的布局的模板数据,而不考虑应用软件 2。当执行应用软件 2 并进行与操作事件相对应的画面显示时,行驶中用 UI API33 仅通过将构成画面的显示要素 (字符串、图像、按钮等) 的一部分应用于该模板,或者替换为该数据中预先准备的简易的字符或字符串 (例如,“执行 abc?”),或者配置与该数据中预先准备的简易的画面操作 (例如,“语音朗读”) 相对应的显示要素,就能生成适合车辆行驶中的行驶中用画面的画面数据。



另外,本发明中,所谓适合车辆行驶中的画面是指,例如为了不分散驾驶员的注意力而对包含与画面操作相关的显示要素的显示内容进行省略和变更后得到的画面。

[0026] 此外,上述模板数据是规定与应用软件 2 无关地构成的画面布局的模板,因而构成画面的显示要素即字符串、图像、按钮等的配置、尺寸、字体、字体大小、字符数等原则上不能改变。

但是,也并非完全固定,以在规定不会分散驾驶员的注意力的范围的规定的限制范围内为条件,可对显示要素的形态进行改变。

例如,在将适合车辆行驶中的情况下的字体大小设为 20 号以上时,行驶中用 UI API33 根据来自应用软件 2 的指示,在从行驶中用画面的模板数据生成画面数据时,以该 20 号为下限来改变字体大小。

[0027] 而且,也可以为应用软件执行环境 3 预先准备多个模板数据,该多个模板数据分别规定适合车辆行驶中的画面结构的多种布局,从而行驶中用 UI API33 能够从这些模板数据中,根据应用软件 2 的指定内容来选择模板数据。

即使这样,各模板数据中所规定的行驶中用画面的布局也无法通过应用软件 2 来改变,因此,由应用软件 2 所指定的画面结构确实是适合车辆行驶中的画面(行驶中用画面)。

此外,具有以下效果:即,对应用软件 2 的开发者来说,通过使用模板数据,能方便地指定行驶中用画面。

[0028] 接着对动作进行说明。

图 8 是表示实施方式 1 所涉及的移动体用信息设备的动作的流程图,详细地示出对应于车辆的停止状态或行驶状态的画面显示。

这里,图 8(a) 示出因执行应用软件 2 而产生的处理,图 8(b) 示出应用软件执行环境 3 中的处理。

[0029] 应用软件执行环境 3 中,控制部 31 在接收到事件后(步骤 ST1a),对接收到的事件的种类进行判定(步骤 ST2a)。

这里,将事件的种类设为来自行驶中判断部 4 的行驶状态变化事件、以及来自操作部 6 的操作事件。

另外,行驶状态变化事件是指表示车辆的行驶状态的变化的事件,表示行驶中的车辆已停止、停止的车辆开始行驶的情况。

此外,操作事件是表示显示部 5 的画面上所显示的按钮的触摸、或按下按键等操作的事件。这里,尤其是指用于利用应用软件 2 进行画面显示的操作。

[0030] 控制部 31 在接收到的事件的种类为“行驶状态变化事件”的情况下(步骤 ST2a:行驶状态变化事件),转移至步骤 ST6a 的处理。

此外,在事件的种类为“操作事件”的情况下(步骤 ST2a:操作事件),控制部 31 经由事件通知部 34 向应用软件执行环境 3 中所执行的应用软件 2 通知操作事件(步骤 ST3a)。

[0031] 应用软件 2 在从应用软件执行环境 3 接收到事件的通知后(步骤 ST1),指定对应于该事件的通常用画面结构(步骤 ST2)。

即,在通知事件后,应用软件 2 调用通常用 UI API32,指定构成与事件内容相对应的通常用画面的显示要素及其显示内容。通常用 UI API32 生成由应用软件 2 指定的通常用画面的画面数据(例如,参照图 2),并将其传输至应用软件执行环境 3 的控制部 31。另外,在

生成通常用画面时,可对构成画面的字符串、图像、按钮等的配置、尺寸、字体、字体大小进行适当地变更。

[0032] 接着,应用软件 2 指定对应于由应用软件执行环境 3 通知的事件的行驶中用画面结构(步骤 ST3)。

即,应用软件 2 调用行驶中用 UI API33,指定构成与事件内容相对应的行驶中用画面的显示要素及其显示内容。

行驶中用 UI API33 基于规定行驶中用画面结构的布局的模板数据、和由应用软件 2 指定的内容,来生成行驶中用画面的画面数据(例如,参照图 5、图 7),并将其传输至应用软件执行环境 3 的控制部 31。由此,行驶中用 UI API33 在通常用 UI API32 生成画面数据后,生成与其相对应的行驶中用画面结构的画面数据,因此,例如,在车辆从停止状态转移至行驶中的状态时,能够迅速地将通常用画面切换成行驶中用画面。

行驶中用 UI API33 在步骤 ST3 的处理结束后,返回至步骤 ST1,每当接收到事件时,则重复进行从步骤 ST1 到步骤 ST3 的处理。

[0033] 控制部 31 接受通常用画面结构(步骤 ST4a),接着,接受行驶中用画面结构(步骤 ST5a)。即,控制部 31 从通常用 UI API32 输入通常用画面的画面数据,从行驶中用 UI API33 输入行驶中用画面的画面数据。然后,控制部 31 对车辆是否处于行驶中进行判定(步骤 ST6a)。该判定是通过参照行驶中判断部 4 所作出的车辆是否处于行驶中的判断结果来进行。另外,在从行驶中判断部 4 接收到行驶中状态变化事件的情况下也进行该处理。

[0034] 在车辆处于停止中的情况下(步骤 ST6a:否),控制部 31 对通常用画面的画面数据进行分析,并根据基于该分析结果的绘制命令来进行通常用画面的绘制处理。显示部 5 输入由控制部 31 生成的绘制数据,并显示通常用画面(步骤 ST7a)。

并且,在车辆处于行驶中的情况下(步骤 ST6a:是),控制部 31 对行驶中用画面的画面数据进行分析,并根据基于该分析结果的绘制命令来进行行驶中用画面的绘制处理。显示部 5 输入由控制部 31 生成的绘制数据,并显示行驶中用画面(步骤 ST8a)。

然后,应用软件执行环境 3 重复进行上述处理。

[0035] 如上所述,根据本实施方式 1,包括:通常用 UI API32,该通常用 UI API32 生成由应用软件 2 指定的画面结构的画面数据;行驶中用 UI API33,该行驶中用 UI API33 基于规定车辆行驶过程中所显示的行驶中用的画面结构的布局的模板数据,来生成由应用软件 2 指定的车辆行驶过程中所显示的行驶中用画面结构的画面数据;以及控制部 31,该控制部 31 设于应用软件执行环境 3,在车辆处于停止时,在显示部 5 中显示由通常用 UI API32 生成的画面数据,在车辆正在行驶时,在显示部 5 中显示由行驶中用 UI API33 生成的画面数据。通过采用这种结构,不论应用软件 2 的动作如何,都能显示适合车辆行驶中的画面。

[0036] 此外,即使应用软件 2 是由车载信息设备厂商以外的第三方开发的应用软件,在车辆处于行驶中的情况下,也仅显示适合行驶中的画面,因而车载信息设备厂商也不需要检查是否显示了不适合行驶中的画面。

另外,在现有技术中,在执行由第三方开发的应用软件时所显示的画面是否适合车辆行驶中这一情况不清楚时,考虑到对其检查所耗费的人力,因而设为只要车辆在行驶中则画面处于非显示状态且不可进行画面操作,但根据上述实施方式 1,能够仅显示图 5、图 7 所示那样的适合行驶中的画面。

[0037] 此外,通过使模板数据中预先包含有简易的画面操作的显示要素,即使在行驶中用画面中,也可以在不分散驾驶员的注意力的范围内进行画面操作,从而能够提高用户的便利性。

[0038] 并且,应用软件 2 的开发者也可以利用行驶中 UI API33 中所定义的行驶中用画面结构,来根据每个应用软件 2 或根据应用软件 2 所执行的每个处理来方便地构建适合行驶中的画面。

[0039] 此外,根据本实施方式 1,应用软件执行环境 3 具有多个模板数据,该多个模板数据分别规定行驶中用画面结构的多个布局,行驶中用 UI API33 基于根据应用软件 2 的指定内容从多个模板数据中选择出的模板数据,来生成行驶中用画面结构的画面数据,因此,能够容易地构建适合车辆行驶过程中的画面数据。

[0040] 并且,根据本实施方式 1,行驶中用 UI API33 根据应用软件 2 的指令,对构成由模板数据规定的画面结构的布局的显示要素进行变更,进而生成行驶中用画面结构的画面数据。

例如,将规定行驶中用画面结构的模板数据中的字符串替换为应用软件 2 指定的字符串,进而生成行驶中用画面的画面数据。

由此,能够构建与应用软件 2 相应的行驶中用画面。另外,即使将上述字符串替换成字符或字符串以外的简易的图像等,也能获得相同的效果。

[0041] 并且,根据本实施方式 1,行驶中用 UI API33 在规定的限制范围内,对根据应用软件 2 的指令来构成基于模板数据生成的行驶中用画面的显示要素的形态进行变更。例如,可在规定为不会分散驾驶员的注意力的规定的限制范围内对显示要素的形态进行变更。由此,能够提高用户的便利性。

[0042] 实施方式 2.

上述实施方式 1 中,示出每次均由应用软件 2 对应用软件执行环境 3 指定通常用画面结构和行驶中用画面结构的情况。

本实施方式 2 中,对以下方式进行说明:即,通过由应用软件执行环境 3 向应用软件 2 通知车辆处于行驶中的情况,使得应用软件 2 仅指定行驶中用画面结构。

[0043] 另外,虽然进行以下处理:即,根据表示车辆处于行驶中的通知,使得应用软件 2 仅指定行驶中用画面结构,但实施方式 2 所涉及的移动体用信息设备的基本结构与实施方式 1 相同。因此,关于实施方式 2 所涉及的移动体用信息设备的结构,参照图 1 所示的车载信息设备 1 的结构。

[0044] 接着对动作进行说明。

图 9 是表示本发明的实施方式 2 所涉及的移动体用信息设备的动作的流程图,详细地示出对应于车辆的停止状态或行驶中状态的画面显示。

这里,图 9(a) 示出因执行应用软件 2 而产生的处理,图 9(b) 示出应用软件执行环境 3 中的处理。

[0045] 应用软件执行环境 3 中,控制部 31 在接收到来自行驶中判断部 4 的行驶状态变化事件、或来自操作部 6 的操作事件后(步骤 ST1c),经由事件通知部 34,将接收到的事件通知给应用软件 2(步骤 ST2c)。

此时,控制部 31 参照由行驶中判断部 4 所作出的车辆是否处于行驶中的判断结果,将

表示车辆行驶状态的数据包含在通知的事件中。然后,控制部 31 在车辆处于停止中时(步骤 ST3c:否),转移至步骤 ST4c 的处理,在车辆处于行驶中时(步骤 ST3c:是),转移至步骤 ST6c 的处理。

[0046] 应用软件 2 在接收到来自应用软件执行环境 3 的事件的通知后(步骤 ST1b),基于该事件中所包含的表示车辆行驶状态的数据,来判定车辆是否处于行驶中(步骤 ST2b)。

这里,若车辆处于停止中(步骤 ST2b:否),则应用软件 2 指定与接收到的事件相对应的通常用画面结构(步骤 ST3b)。

即,与上述实施方式 1 相同,应用软件 2 调用通常用 UI API32,指定构成与事件内容相对应的通常用画面的显示要素及其显示内容。通常用 UI API32 生成由应用软件 2 指定的通常用画面的画面数据,并将其传输至应用软件执行环境 3 的控制部 31。

[0047] 控制部 31 接受通常用画面结构(步骤 ST4c)。即,控制部 31 从通常用 UI API32 输入通常用画面的画面数据。

然后,控制部 31 对通常用画面的画面数据进行分析,并根据基于该分析结果的绘制命令来进行通常用画面的绘制处理。显示部 5 输入由控制部 31 生成的绘制数据,并显示通常用画面(步骤 ST5c)。

[0048] 另一方面,若车辆处于行驶中(步骤 ST2b:是),则应用软件 2 指定与接收到的事件相对应的行驶中用画面结构(步骤 ST4b)。

即,与上述实施方式 1 相同,应用软件 2 调用行驶中用 UI API33,指定构成与事件内容相对应的行驶中用画面的显示要素及其显示内容。行驶中用 UI API33 基于对行驶中用画面结构的布局进行了规定的模板数据和由应用软件 2 指定的内容,生成行驶中用画面的画面数据,并将其传输至应用软件执行环境 3 的控制部 31。

[0049] 接着,控制部 31 接受行驶中用画面结构(步骤 ST6c)。

即,控制部 31 从行驶中用 UI API33 输入行驶中用画面的画面数据。此时,控制部 31 对是否从行驶中用 UI API33 正常接收到画面数据进行判定(步骤 ST7c)。这里,以是否能够以可分析画面数据的状态进行接收,或者以是否能够在规定的接受时间内进行接收作为判定基准,来判定是否正常接受。

[0050] 在判定为正常接受到画面数据的情况下(步骤 ST7c:是),控制部 31 对该画面数据进行分析,并根据基于该分析结果的绘制命令来进行行驶中用画面的绘制处理。显示部 5 输入由控制部 31 生成的绘制数据,并显示行驶中用画面(步骤 ST8c)。

然后,应用软件执行环境 3 重复进行上述处理。

[0051] 此外,在由于未能以可分析画面数据的状态进行接收、或者未能在规定的接受时间内进行接收、而判定为未正常接受到画面数据的情况下(步骤 ST7c:否或超时),控制部 31 对应用软件执行环境 3 中预先准备的既定的行驶中用画面数据进行分析,并根据基于该分析结果的绘制命令,来进行行驶中用画面的绘制处理。显示部 5 输入由控制部 31 生成的绘制数据,并显示既定的行驶中用画面(步骤 ST9c)。

然后,应用软件执行环境 3 重复进行上述处理。

另外,既定的行驶中用画面数据是指与应用软件 2 和对应于事件的处理无关的画面数据,该画面数据表示根据车辆处于行驶中的情况简化了显示内容的画面。

[0052] 如上所述,根据本实施方式 2,通常用 UI API32 在车辆处于停止时生成通常用画

面的画面数据,行驶中用 UI API33 在车辆正在行驶时生成行驶中用画面的画面数据。

由此,应用软件 2 根据车辆处于停止还是处于行驶中,利用通常用 UI API32 和行驶中用 UI API33,来指定通常用画面结构和行驶中用画面结构中的一个,由此能够减少应用软件 2 的处理量。

另外,这种情况下,在车辆处于停止中和车辆处于行驶中能够进行不同画面的转换。

[0053] 实施方式 3.

上述实施方式 1,2 中,示出以下情况:在显示部 5 上进行画面显示时,生成通常用画面和行驶中用画面中的至少一个的画面数据,并显示与该至少一个的画面数据相关的画面。

本实施方式 3 对以下方式进行说明:即,设有保存分析画面数据而得到的绘制数据的离屏缓存,分别生成通常用画面和行驶中用画面的绘制数据,并将其预先绘制在离屏缓存中,根据车辆的行驶状态来显示离屏缓存的各画面的绘制数据。

[0054] 另外,虽然进行以下处理动作:即,在离屏缓存中绘制通常用画面和行驶中用画面来进行画面显示,但实施方式 3 所涉及的移动体信息设备的基本结构与上述实施方式 1 相同。因此,关于实施方式 3 所涉及的移动体用信息设备的结构,参照图 1 所示的车载信息设备 1 的结构。

[0055] 接着对动作进行说明。

图 10 是表示本发明的实施方式 3 所涉及的移动体用信息设备的动作的流程图,详细地示出对应于车辆的停止状态或行驶状态的画面显示。

这里,图 10(a) 示出因执行应用软件 2 而产生的处理,图 10(b) 示出应用软件执行环境 3 中的处理。

[0056] 应用软件执行环境 3 中,控制部 31 在接收到事件后(步骤 ST1e),与上述实施方式 1 相同,对接收到的事件的种类进行判定(步骤 ST2e)。这里,事件的种类设为来自行驶中判断部 4 的行驶状态变化事件、以及来自操作部 6 的操作事件。

[0057] 控制部 31 在接收到的事件的种类为“行驶状态变化事件”的情况下(步骤 ST2e:行驶状态变化事件),转移至步骤 ST8e 的处理。

此外,在事件的种类为“操作事件”的情况下(步骤 ST2e:操作事件),控制部 31 经由事件通知部 34 向应用软件执行环境 3 中所执行的应用软件 2 通知操作事件(步骤 ST3e)。

[0058] 应用软件 2 在从应用软件执行环境 3 接收到事件的通知后(步骤 ST1d),指定对应于接收到的事件的通常用画面结构(步骤 ST2d)。即,与上述实施方式 1 相同,应用软件 2 调用通常用 UI API32,指定构成与事件内容相对应的通常用画面的显示要素及其显示内容。通常用 UI API32 生成由应用软件 2 指定的通常用画面的画面数据,并将其传输至应用软件执行环境 3 的控制部 31。

[0059] 接着,应用软件 2 指定对应于由应用软件执行环境 3 通知的事件的行驶中用画面结构(步骤 ST3d)。

即,应用软件 2 调用行驶中用 UI API33,指定构成与事件内容相对应的行驶中用画面的显示要素及其显示内容。

行驶中用 UI API33 基于对行驶中用画面结构的布局进行了规定的模板数据和由应用软件 2 指定的内容,生成行驶中用画面的画面数据,并将其传输至应用软件执行环境 3 的控制部 31。

行驶中用 UI API33 在步骤 ST3d 的处理结束后,返回至步骤 ST1d,每当接收到事件时,则重复进行从步骤 ST1d 到步骤 ST3d 的处理。

[0060] 控制部 31 接受通常用画面结构 (步骤 ST4e),接着,接受行驶中用画面结构 (步骤 ST5e)。即,控制部 31 从通常用 UI API32 输入通常用画面的画面数据,从行驶中用 UI API33 输入行驶中用画面的画面数据。

接着,控制部 31 对通常用画面的画面数据进行分析,并根据基于该分析结果的绘制命令来生成通常用画面的绘制数据,并将其绘制 (保存) 在离屏缓存中 (步骤 ST6e)。

接着,控制部 31 对行驶中用画面的画面数据进行分析,并根据基于该分析结果的绘制命令来生成行驶中用画面的绘制数据,并以位于与通常用画面的绘制数据不同的显示层的方式来将其绘制 (保存) 在离屏缓存中 (步骤 ST7e)。

[0061] 然后,控制部 31 对车辆是否处于行驶中进行判定 (步骤 ST8e)。该判定与上述实施方式 1 相同,通过参照行驶中判断部 4 所作出的车辆是否处于行驶中的判断结果来进行。

在车辆处于停止中时 (步骤 ST8e :否),控制部 31 对显示部 5 进行控制,使得显示部 5 显示绘制在离屏缓存中的通常用画面的绘制数据。由此,显示部 5 显示绘制在离屏缓存中的通常用画面 (步骤 ST9e)。

此外,在车辆处于行驶中时 (步骤 ST8e :是),控制部 31 对显示部 5 进行控制,使得显示部 5 切换至绘制在离屏缓存中的行驶中用画面的绘制数据来进行显示。由此,显示部 5 显示绘制在离屏缓存中的行驶中用画面 (步骤 ST10e)。

[0062] 如上所述,根据本实施方式 3,包括保存有对画面数据进行绘制处理而得到的绘制数据的离屏缓存,控制部 31 以使得由通常用 UI API32 生成的画面数据的绘制数据、和由行驶中用 UI API33 生成的画面数据的绘制数据位于不同的显示层的方式,来将其保存在离屏缓存中,根据车辆是否正在行驶来切换离屏缓存中所保存的各绘制数据,并在显示部 5 中进行显示。根据这种结构,在车辆状态发生变化时,仅通过切换离屏缓存中所保存的绘制数据,就能显示通常用画面或行驶中用画面,从而能够在短时间内进行画面显示的切换。

[0063] 另外,上述实施方式 3 中,示出了切换通常用画面和行驶中用画面并进行显示的情况,但是,例如,在车辆处于行驶中的情况下,也可以如图 11 所示那样,在通常用画面的层上重叠行驶中用画面的层来进行显示。在这种情况下,为了提高设计性,可以使下层画面的一部分透过上层或半透过上层来进行显示。

[0064] 实施方式 4.

上述实施方式 1 ~ 3 中,示出了具备用于指定通常用画面结构的通常用 UI API32 和用于指定行驶中用画面结构的行驶中用 UI API33 的结构。

本实施方式 4 对以下方式进行描述 :即,仅具备通常用 UI API32 以作为用于指定画面结构的 API,在车辆处于行驶中的情况下,根据通常用 UI API32 生成的通常用画面的画面数据,来生成行驶中用画面的画面数据。

[0065] 图 12 是表示本发明的实施方式 4 所涉及的移动体用信息设备的结构的框图,示出了将实施方式 4 所涉及的移动体用信息设备应用于车载信息设备的情况。图 12 所示的车载信息设备 1A 中设置有执行应用软件 2 的应用软件执行环境 3A、行驶中判断部 4、显示部 5 以及操作部 6。

应用软件执行环境 3A 是执行应用软件 2 的执行环境,包括控制部 31、通常用 UI API32、

事件通知部 34 以及行驶中用 UI 生成部 35。即,应用软件执行环境 3A 相当于在图 1 所示的车载信息设备 1 的应用软件执行环境 3 中设置行驶中用 UI 生成部 35 来代替行驶中用 UI API33。

行驶中用 UI 生成部 35 根据规定的规则,从由通常用 UI API32 生成的通常用画面的画面数据来生成行驶中用画面的画面数据。另外,图 12 中,对于与图 1 相同的结构要素标注相同的符号,并省略说明。

[0066] 接着对动作进行说明。

图 13 是表示实施方式 4 所涉及的移动体用信息设备的动作的流程图,详细地示出对应于车辆的停止或行驶的车载信息设备 1A 的画面显示。

这里,图 13(a) 示出因执行应用软件 2 而产生的处理,图 13(b) 示出应用软件执行环境 3A 中的处理。

应用软件执行环境 3A 中,控制部 31 在接收到事件后(步骤 ST1g),与上述实施方式 1 相同,对接收到的事件的种类进行判定(步骤 ST2g)。这里,事件的种类设为来自行驶中判断部 4 的行驶状态变化事件、以及来自操作部 6 的操作事件。

[0067] 控制部 31 在接收到的事件的种类为“行驶状态变化事件”的情况下(步骤 ST2g:行驶状态变化事件),转移至步骤 ST6g 的处理。

此外,在事件的种类为“操作事件”的情况下(步骤 ST2g:操作事件),控制部 31 经由事件通知部 34 向应用软件执行环境 3A 中所执行的应用软件 2 通知操作事件(步骤 ST3g)。

[0068] 应用软件 2 在从应用软件执行环境 3A 接收到事件的通知后(步骤 ST1f),指定对应于该事件的通常用画面结构(步骤 ST2f)。即,与上述实施方式 1 相同,应用软件 2 调用通常用 UI API32,指定构成与事件内容相对应的通常用画面的显示要素及其显示内容。通常用 UI API32 生成由应用软件 2 指定的通常用画面的画面数据,并将其传输至应用软件执行环境 3A 的控制部 31。控制部 31 接受通常用画面结构(步骤 ST4g)。即,控制部 31 从通常用 UI API32 输入通常用画面的画面数据。

[0069] 接着,行驶中用 UI 生成部 35 从控制部 31 输入通常用画面的画面数据,根据该画面数据,基于规定的规则,自动地生成行驶中用画面的画面数据(步骤 ST5g)。

例如,设定下述(1)~(3)的规则。

(1) 选择“template-A”作为行驶中用画面的模板。

(2) 提取出通常用画面的画面数据中最开始的字符串,将其替换作为行驶中用画面的模板中的“msg1”所规定的页标头的字符串。

(3) 提取出通常用画面的画面数据中的、从开头起的两个 button 要素,替换行驶中用画面的模板中的按钮的字符串。

[0070] 图 14 是根据图 2 所示的通常用画面的画面数据、基于上述(1)~(3)的规则所生成的行驶中用画面的画面数据。

如图 14 所示,行驶中用 UI 生成部 35 选择“template-A”作为行驶中用画面的模板。

接着,行驶中用 UI 生成部 35 提取出通常用画面的画面数据中最开始的字符串即“新闻:标题”(参照图 2),替换作为上述模板中的“msg1”所规定的页标头中所描述的字符串。

接着,行驶中用 UI 生成部 35 提取出通常用画面的画面数据中的、从开头起按顺序排列的两个 button 要素即“返回”和“语音朗读”,将行驶中用画面的模板中的按钮中描述的字

字符串替换为“返回”和“语音朗读”。

由此,生成与图 5 相同的行驶中用画面的画面数据。

[0071] 返回图 13 的说明。

控制部 31 在输入通常用画面的画面数据、和由行驶中用 UI 生成部 35 生成的行驶中用画面的画面数据后,对车辆是否处于行驶中进行判定(步骤 ST6g)。该判定通过参照行驶中判断部 4 所作出的车辆是否处于行驶中的判断结果来进行。

在车辆处于停止中的情况下(步骤 ST6g:否),控制部 31 对通常用画面的画面数据进行分析,并根据基于该分析结果的绘制命令来进行通常用画面的绘制处理。显示部 5 输入由控制部 31 生成的绘制数据,并显示通常用画面(步骤 ST7g)。

[0072] 另一方面,在车辆处于行驶中的情况下(步骤 ST6g:是),控制部 31 对行驶中用画面的画面数据进行分析,并根据基于该分析结果的绘制命令来进行行驶中用画面的绘制处理。显示部 5 输入由控制部 31 生成的绘制数据,并显示行驶中用画面(步骤 ST8g)。

然后,应用软件执行环境 3A 重复进行上述处理。

[0073] 如上所述,根据本实施方式 4,由于具备根据通常用画面的画面数据生成行驶中用画面的画面数据的行驶中用 UI 生成部 35,因此,仅通过由应用软件 2 指定通常用画面结构,就能够同时指定行驶中用画面结构。

此外,行驶中用 UI 生成部 35 在通常用 UI API32 生成画面数据后,生成与其相对应的行驶中用画面结构的画面数据,因此在车辆的状态(停止或行驶)发生变化时,能迅速地切换至对应于变化后的车辆状态的画面。

[0074] 此外,上述实施方式 4 中,示出以下情况:在步骤 ST5g 中,行驶中用 UI 生成部 35 根据通常用画面的画面数据生成行驶中用画面的画面数据,然后在步骤 ST6g 中,当车辆处于行驶中时,利用基于行驶中用画面的画面数据的绘制数据,在显示部 5 中显示行驶中用画面。

本发明并不限于上述处理的流程,也可以按下述方式进行处理:即,在车辆是否处于行驶中的判定结果出来之前,行驶中用 UI 生成部 35 不根据通常用画面的画面数据生成行驶中用画面的画面数据,而仅在上述判定为车辆处于行驶中的情况下,才根据通常用画面的画面数据生成行驶中用画面的画面数据,并利用基于行驶中用画面的画面数据的绘制数据,在显示部 5 中显示行驶中用画面。

[0075] 并且,上述实施方式 4 中,优选为当显示部 5 上显示图像、动画、影像等时,在车辆处于行驶中的情况下,将动画或影像等视频变更为静止图像来进行显示。

图 15 是表示以 HTML 格式呈现车辆处于停止时的画面结构的画面数据的一个示例的图,示出显示要素中包含有动画图像的通常用画面的画面数据。此外,图 16 是表示基于图 15 的画面数据进行显示的画面的图。图 15 中,设为由“img”要素来指定动画要素。此外,在图 16 中,由“img”要素指定的动画 a 显示在描述有“ABC 获胜!”、“日元进一步走高”、“DEF 公司和 GHI 公司合作”的矩形的右侧。

[0076] 行驶中用 UI 生成部 35 根据下述(1A)~(4A)的规则,从图 15 所示的通常用画面的画面数据生成行驶中用画面的画面数据。

(1A) 选择“template-C”作为行驶中用画面的模板。

(2A) 提取出通常用画面的画面数据中最开始的字符串,将其替换作为行驶中用画面的



模板中的“msg1”规定的页标头的字符串。

(3A) 提取出通常用画面的画面数据中从开头起的两个 button 要素, 替换行驶中用画面的模板中的按钮的字符串。

(4A) 提取出通常用画面的画面数据中最开始的动画, 利用将该动画转换成静止图像而得到的内容替换“img”要素。

[0077] 图 17 示出行驶中用 UI 生成部 35 根据上述 (1A) ~ (4A) 的规则, 由图 15 的画面数据生成的行驶中用画面的画面数据。此外, 图 18 是表示基于图 17 的画面数据进行显示的画面的图。

图 17 中的“animation-fixed.gif”是将图 15 的通常用画面的画面数据中、由“animation.gif”表示的动画转换成静止图像而得到的。对于动画向静止图像的转换, 由行驶中用 UI 生成部 35 来实施, 例如, 提取出动画中规定的帧图像 ( 最开始的帧等 ) 作为静止图像。

[0078] 根据基于图 17 的画面数据生成的绘制数据, 在显示部 5 中显示图 18 所示的行驶中用画面。如图 18 所示, 在图 16 的画面中记载有动画 a 的位置, 记载了由动画 a 转换得到的静止图像 b。

如上所示, 在根据通常用画面的画面数据生成行驶中用画面的画面数据时, 通过将动画或视频转换成静止图像, 从而能够显示适合车辆行驶中的画面。

[0079] 并且, 在上述实施方式 4 中, 也可以采用以下方式: 即, 通常用 UI API32 在通常用画面的画面数据中预先包含有构成行驶中用画面的信息作为附带信息, 行驶中用 UI 生成部 35 根据该附带信息生成行驶中用画面的画面数据。

图 19 是表示包含有构成行驶中用画面的信息的通常用画面的画面数据的图。图 19 所示的画面数据是在实施方式 1 所示的图 2 的画面数据中添加“running-ui type”要素和“running-param”属性而得到的。这里, “running-ui type”要素表示由图 19 的画面数据生成的行驶中用画面的画面数据所使用的模板数据。

此外, “running-param”属性表示由上述通常用画面的画面数据生成的行驶中用画面的画面数据中的“text”要素中所记载的字符串。

行驶中用 UI 生成部 35 通过对图 19 的画面数据中包含的构成行驶中用画面的信息即“running-ui type”要素和“running-param”属性的内容进行组合, 从而能够生成行驶中用画面的画面数据。

由图 19 的画面数据可生成与图 4 所示的行驶中用画面的画面数据相同的画面数据。

[0080] 并且, 在上述实施方式 4 中, 设有保存对画面数据进行绘制处理而得到的绘制数据的离屏缓存, 控制部 31 以使得由通常用 UI API32 生成的画面数据的绘制数据、和由行驶中用 UI API35 生成的画面数据的绘制数据位于不同的显示层的方式, 来将其保存在离屏缓存中, 根据车辆是否正在行驶来切换离屏缓存中所保存的各绘制数据, 并在显示部 5 中进行显示。根据这种结构, 也能与上述实施方式 4 相同, 在车辆的状态发生变化时, 仅通过切换离屏缓存中所保存的绘制数据, 就能显示通常用画面或行驶中用画面, 从而能够在短时间内进行画面显示的切换。

[0081] 实施方式 5.

图 20 是表示本发明的实施方式 5 所涉及的移动体用信息设备的结构的框图, 示出了将

实施方式 5 所涉及的移动体用信息设备应用于车载信息设备的情况。图 20 所示的车载信息设备 1B 中设置有执行应用软件 2 的应用软件执行环境 3B、行驶中判断部 4、显示部 5、操作部 6 以及语音操作部 7。

此外,应用软件执行环境 3B 是执行应用软件 2 的执行环境,包括控制部 31A、通常用 UI API32、行驶中用 UI API33、以及事件通知部 34。

[0082] 语音操作部 7 对用户发出的语音进行识别,并将该识别结果作为语音事件通知给应用软件执行环境 3B 的控制部 31A。这里,由控制部 31A 在语音操作部 7 中预先登记命令字符串,在发出与该命令字符串一致或相似的语音的情况下,判定为发生语音事件。

另外,图 20 中,对与图 1 相同的结构要素标注相同的符号,并省略说明。

[0083] 接着对动作进行说明。

图 21 是表示实施方式 5 所涉及的移动体用信息设备的动作的流程图,详细地示出对应于车辆的停止或行驶的车载信息设备 1B 的画面显示。

这里,图 21 (a) 示出因执行应用软件 2 而产生的处理,图 21 (b) 示出应用软件执行环境 3B 中的处理。

应用软件执行环境 3B 中,控制部 31A 在接收到事件后(步骤 ST1i),对接收到的事件种类进行判定(步骤 ST2i)。

这里,事件种类设为来自行驶中判断部 4 的行驶状态变化事件、来自操作部 6 的操作事件、以及来自语音操作部 7 的语音事件。

[0084] 控制部 31A 在接收到的事件种类为“行驶状态变化事件”的情况下(步骤 ST2i: 行驶状态变化事件),转移至步骤 ST6i 的处理。

此外,在事件种类为“操作事件”或“语音事件”的情况下(步骤 ST2i: 操作事件或语音事件),控制部 31A 经由事件通知部 34 向应用软件执行环境 3B 中所执行的应用软件 2 通知该事件(步骤 ST3i)。

[0085] 应用软件 2 在从应用软件执行环境 3B 接收到事件的通知后(步骤 ST1h),指定对应于该事件的通常用画面结构(步骤 ST2h)。即,与上述实施方式 1 相同,应用软件 2 调用通常用 UI API32,指定构成与事件内容相对应的通常画面的显示要素及其显示内容。通常用 UI API32 生成由应用软件 2 指定的通常画面的画面数据,并将其传输至应用软件执行环境 3B 的控制部 31A。

[0086] 接着,应用软件 2 指定由应用软件执行环境 3B 通知的事件所对应的行驶中用画面结构(步骤 ST3h)。

即,应用软件 2 调用行驶中用 UI API33,指定构成与事件内容相对应的行驶中用画面的显示要素及其显示内容。行驶中用 UI API33 基于对行驶中用画面结构的布局进行了规定的模板数据和由应用软件 2 指定的内容,生成行驶中用画面的画面数据,并将其传输至应用软件执行环境 3B 的控制部 31A。

[0087] 另外,由于语音操作是不需要手动操作的适合车辆行驶中的操作,因此本实施方式 5 所涉及的行驶中用 UI API33 将与接收到的事件内容相关的操作的语音命令编入到行驶中用画面的画面数据中。

行驶中用 UI API33 在步骤 ST3h 的处理结束后,返回至步骤 ST1h,每当接收到事件,则重复进行从步骤 ST1h 到步骤 ST3h 的处理。

[0088] 控制部 31A 接受通常用画面结构 (步骤 ST4i),接着,接受行驶中用画面结构 (步骤 ST5i)。即,控制部 31A 从通常用 UI API32 输入通常用画面的画面数据,从行驶中用 UI API33 输入行驶中用画面的画面数据。

然后,控制部 31A 对车辆是否处于行驶中进行判定 (步骤 ST6i)。该判定通过参照行驶中判断部 4 所作出的车辆是否处于行驶中的判断结果来进行。

[0089] 在车辆处于停止中的情况下 (步骤 ST6i:否),控制部 31A 对通常用画面的画面数据进行分析,并根据基于该分析结果的绘制命令来进行通常用画面的绘制处理。显示部 5 输入由控制部 31A 生成的绘制数据,并显示通常用画面 (步骤 ST7i)。然后,应用软件执行环境 3B 重复进行上述处理。

[0090] 另一方面,在车辆处于行驶中的情况下 (步骤 ST6i:是),控制部 31A 对行驶中用画面的画面数据进行分析,并根据基于该分析结果的绘制命令来进行行驶中用画面的绘制处理。显示部 5 输入由控制部 31A 生成的绘制数据,并显示行驶中用画面 (步骤 ST8i)。

接着,控制部 31A 将行驶中用画面的画面数据中包含的语音命令登记到语音操作部 7 (步骤 ST9i)。

[0091] 图 22 是表示编有语音命令的行驶中用画面的画面数据的图。

图 22 的画面数据是在图 4 所示的画面数据中添加两个表示语音命令的“speech”要素而得到的。步骤 ST9i 中,控制部 31A 将“speech”要素中所记载的“返回”和“语音朗读”这样的语音命令登记到语音操作部 7 中。另外,基于图 22 的画面数据来显示的行驶中用画面与图 5 相同。

[0092] 在显示部 5 中显示上述行驶中用画面时,若发出与上述语音命令相一致或相类似的语音,则语音操作部 7 向应用软件执行环境 3B 的控制部 31A 通知语音事件。控制部 31A 在接收到来自语音操作部 7 的语音事件后,经由步骤 ST 事件通知部 34,将该语音事件通知给应用软件 2。

[0093] 如上所述,根据本实施方式 5,由于包括语音操作部 7,该语音操作部 7 对用户发出的语音进行识别,并在识别结果与由控制部 31A 登记的语音命令相一致或相类似的情况下,将该识别结果作为语音事件通知给控制部 31A,行驶中用 UI API33 生成编入有语音命令的行驶中用画面结构的画面数据,因此,在行驶中画面中能够利用语音识别进行操作。

[0094] 另外,上述实施方式 5 中,示出了在上述实施方式 1~3 的结构中添加语音操作部 7 的情况,但也可以在上述实施方式 4 的结构中添加语音操作部 7。

在这种情况下,行驶中用 UI 生成部 35 在根据通常用画面的画面数据生成行驶中用画面的画面数据时,将语音命令编入行驶中用画面的画面数据中。

由此也能够获得与上述相同的效果。

[0095] 此外,在上述实施方式 1~5 中,示出以 HTML 格式或 XML 格式指定画面结构的 API,但也可以利用其他语言或方法来指定画面结构。例如,也可以是使用 Java (注册商标) 语言的类 (class)、方法 (method) 的 API。

[0096] 并且,在上述实施方式 1~5 中,示出在车辆处于行驶中时显示部 5 中显示行驶中用画面的情况,但对于车辆的用于副驾驶、后座等的多个显示部中的主要由驾驶员进行视觉确认的显示部以外的显示部,即使车辆在行驶中,也可以显示通常用画面,而不切换为行驶中用画面。

例如,控制部 31 基于分别对多个显示部进行识别的识别信息,确定主要由驾驶员进行视觉确认的显示部 5,对于该显示部 5,根据车辆是否处于行驶中来切换通常用画面和行驶中用画面,而对于该显示部 5 以外的显示部,即使车辆在行驶中,也仍然显示通常用画面,而不切换为行驶中用画面。

[0097] 上述实施方式 1~5 中,示出将本发明所涉及的移动体用信息设备应用于车载信息设备的情况,但也可以应用于车辆以外的搭载于铁路、船舶或飞机的移动体用信息设备,还可以应用于由人携带到车辆内进行使用的便携式信息终端、例如 PND (Portable Navigation Device:便携式导航装置)。

[0098] 另外,本发明可以在其发明的范围内对各实施方式进行自由组合,或对各实施方式的任意结构要素进行变形、或省略各实施方式中任意的结构要素。

#### 工业上的实用性

[0099] 本发明所涉及的移动体用信息设备能够显示分别适合移动体处于停止中的情况和处于移动中的情况的画面,因此,适用于在车辆行驶中操作内容受到限制的导航装置等车辆信息设备。

#### 标号说明

- [0100] 1, 1A, 1B 车载信息设备
- 2 应用软件
- 3, 3A, 3B 应用软件执行环境
- 4 行驶中判断部
- 5 显示部
- 6 操作部
- 7 语音操作部
- 31, 31A 控制部
- 32 通常用 UI API
- 33 行驶中用 UI API
- 34 事件通知部
- 35 行驶中用 UI 生成部

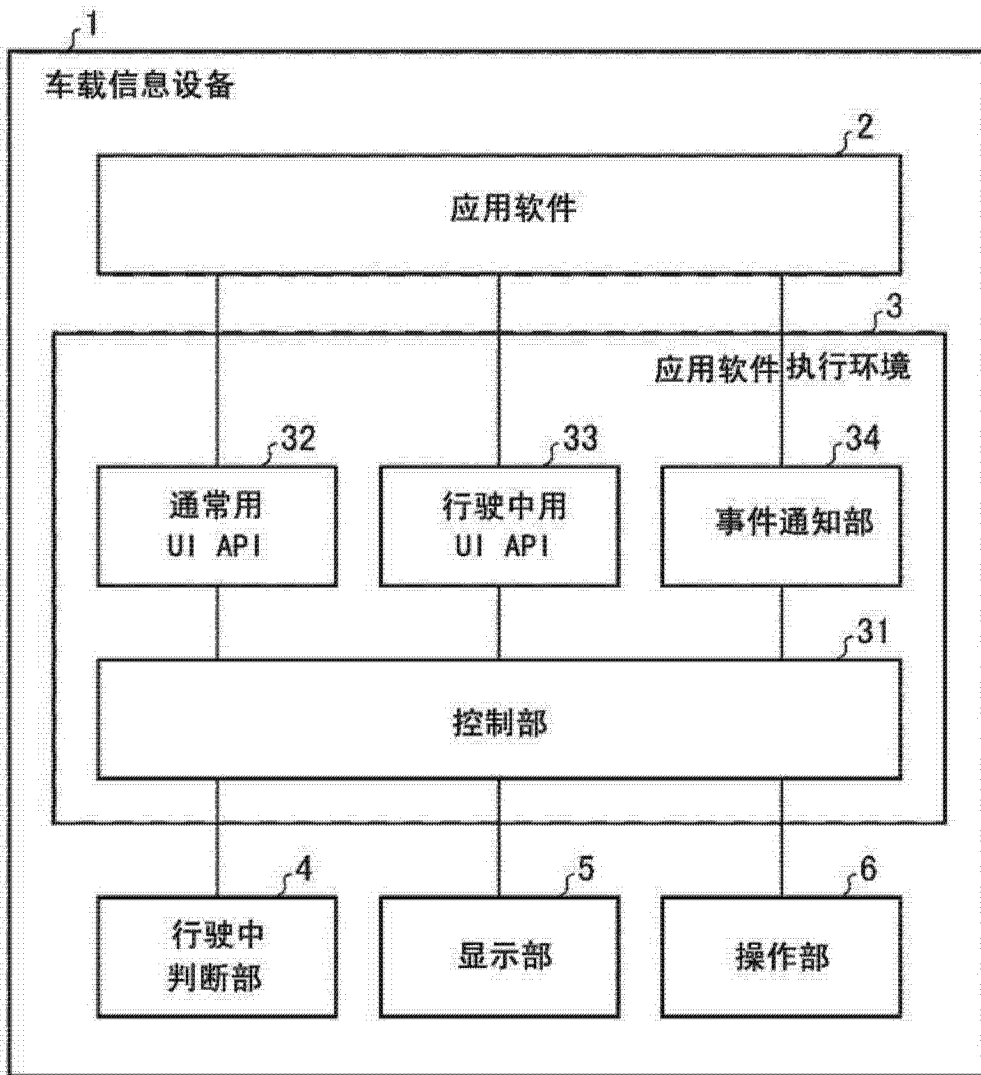


图 1

```
<!DOCTYPE html>
<html>
<style type="text/css">
<!--
body, div, list {
padding: 2px;
margin: 2px;
border: solid 1px Black;
}
button {
width: 90px;
height: 2em;
}
.pagetitle {
background: LightGray;
}
-->
</style>
<body>
<div class="pagetitle">.新闻：标题</div>
<div class="list">
<div class="heading" id="heading1">ABC获胜!</div>
<div class="heading" id="heading2">日元进一步走高</div>
<div class="heading" id="heading3">DEF公司和GHI公司合作</div>
</div>
<button>返回</button>
<button>语音朗读</button>
<button>前一页</button>
<button>后一页</button>
</body>
</html>
```

图 2

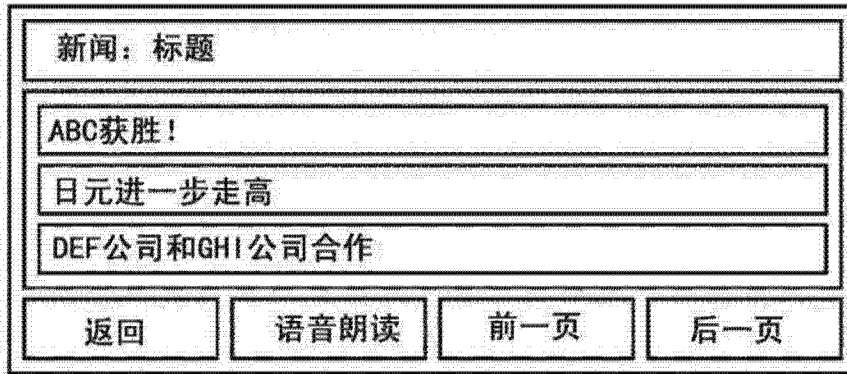


图 3

```
<!DOCTYPE running-ui>  
<running-ui type="template-A">  
<text id="msg1">新闻：标题 </text>  
<text id="btn2">语音朗读 </text>  
</running-ui>
```

图 4

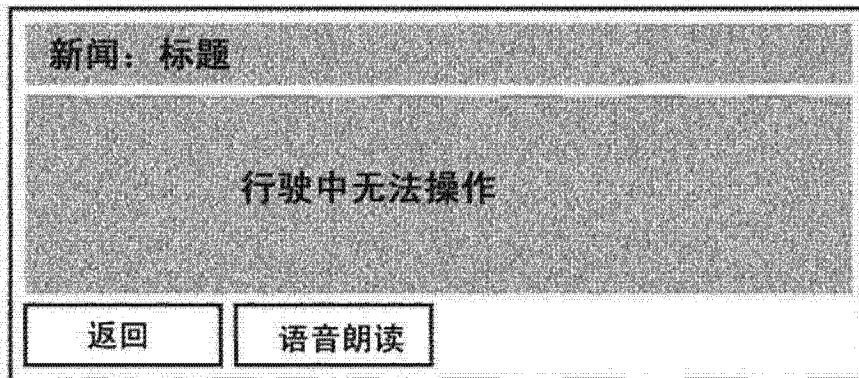


图 5

```
<!DOCTYPE running-ui>  
<running-ui type="template-B">  
<text id="msg1">执行abc? </text>  
</running-ui>
```

图 6

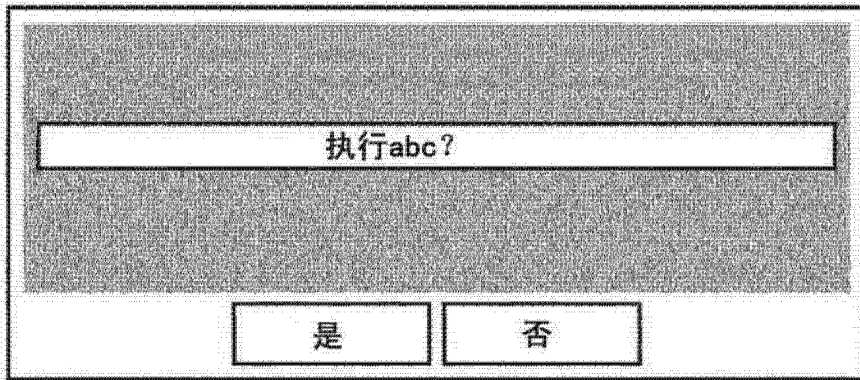


图 7



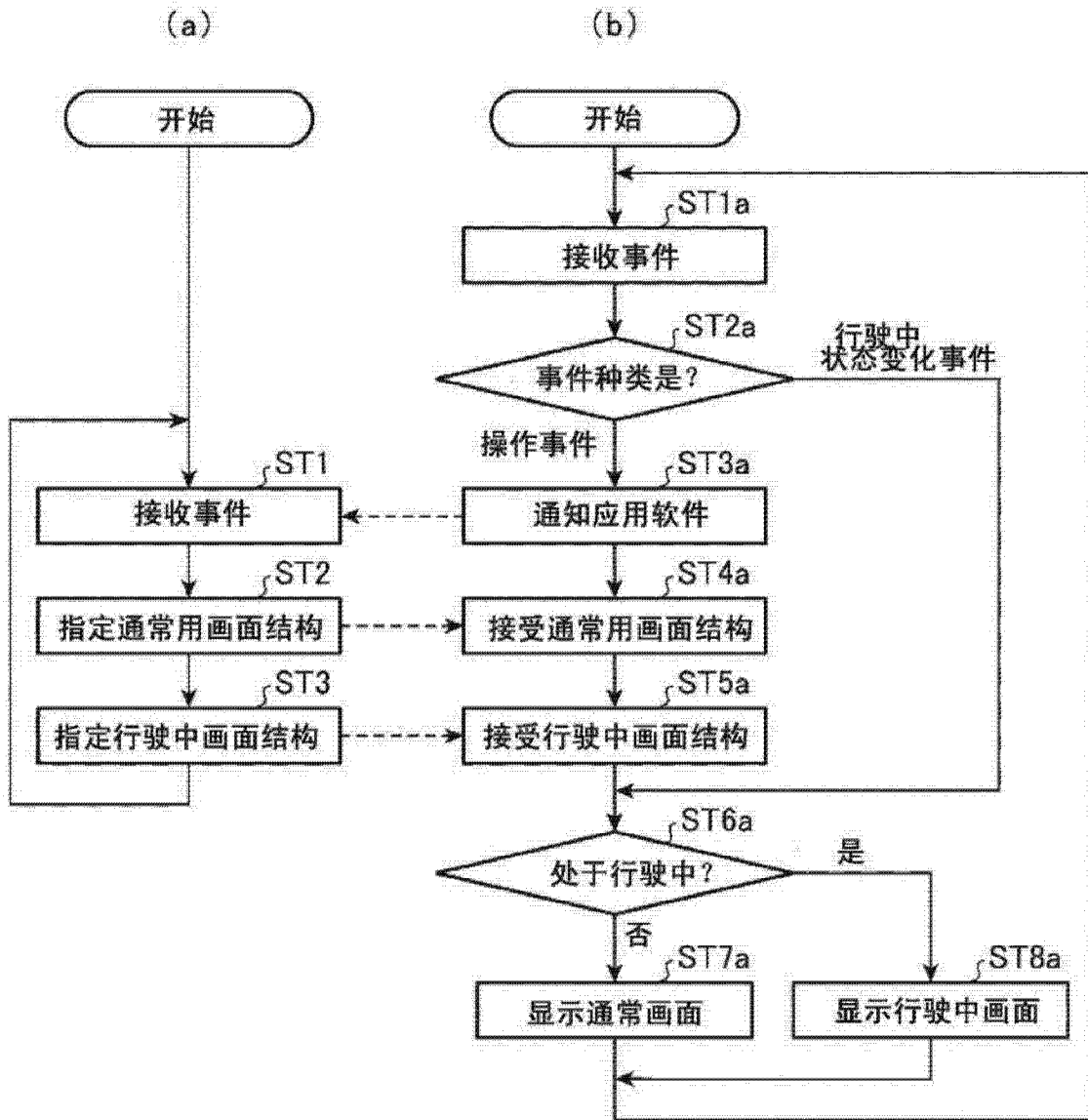


图 8

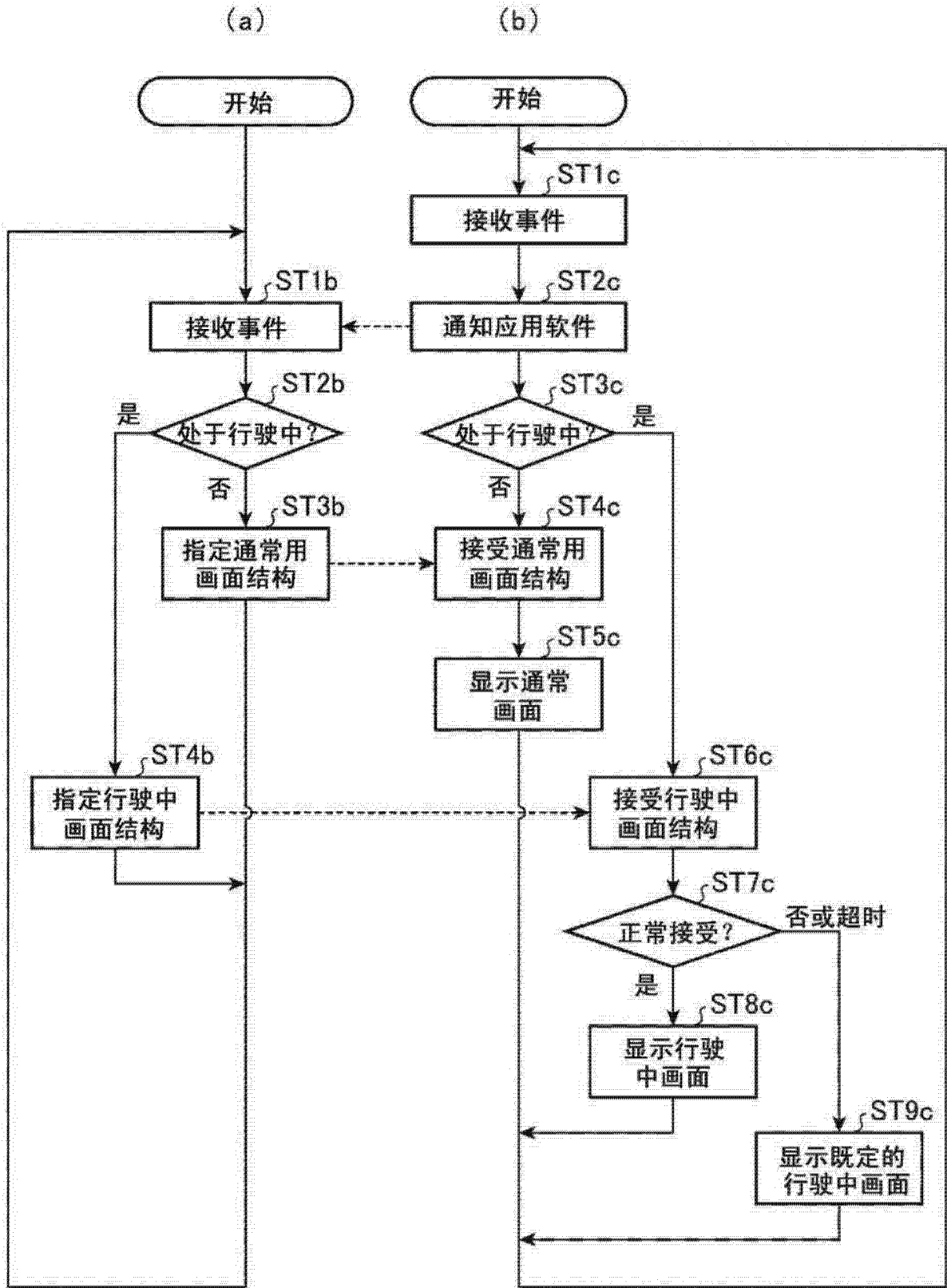


图 9

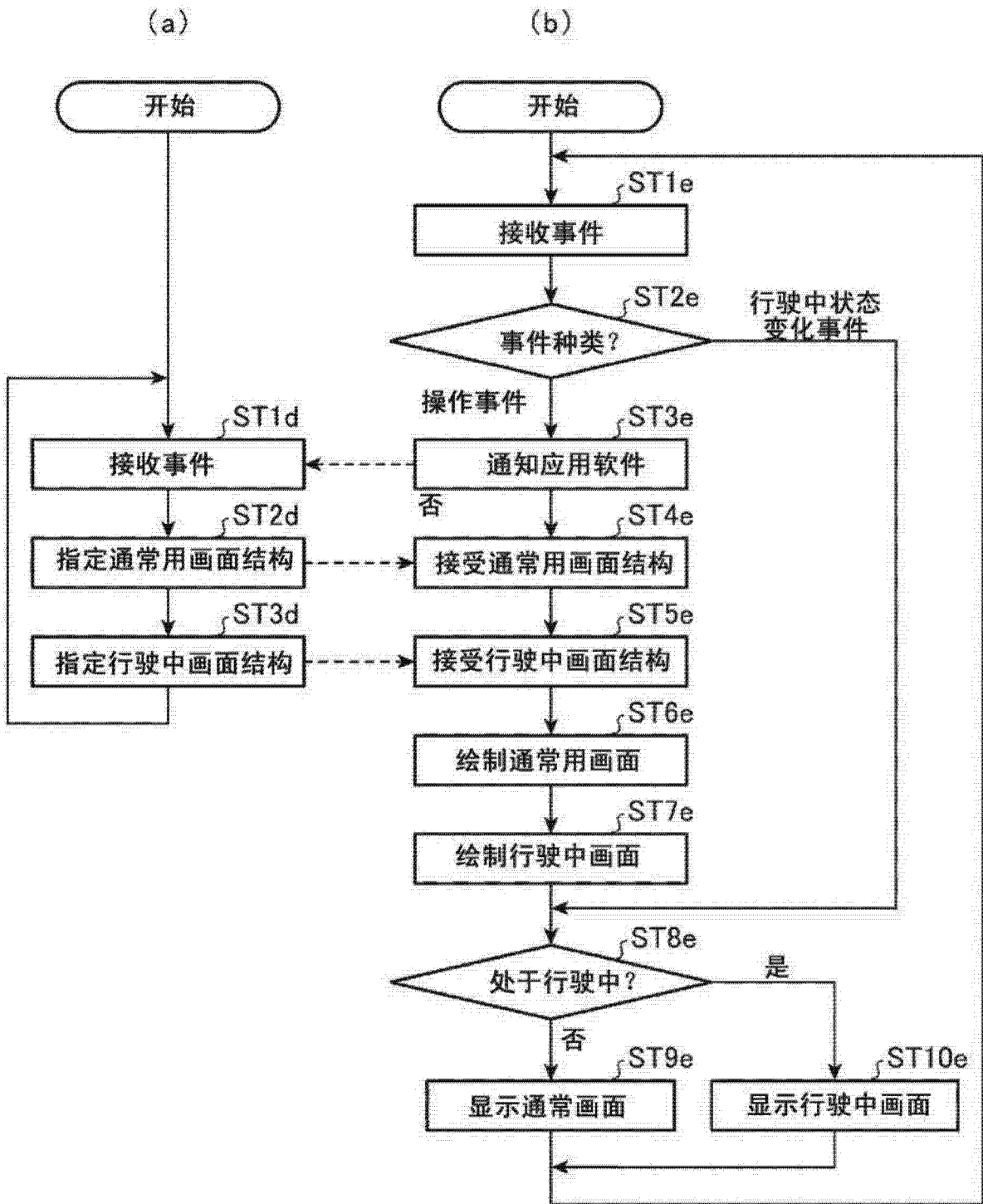


图 10



图 11

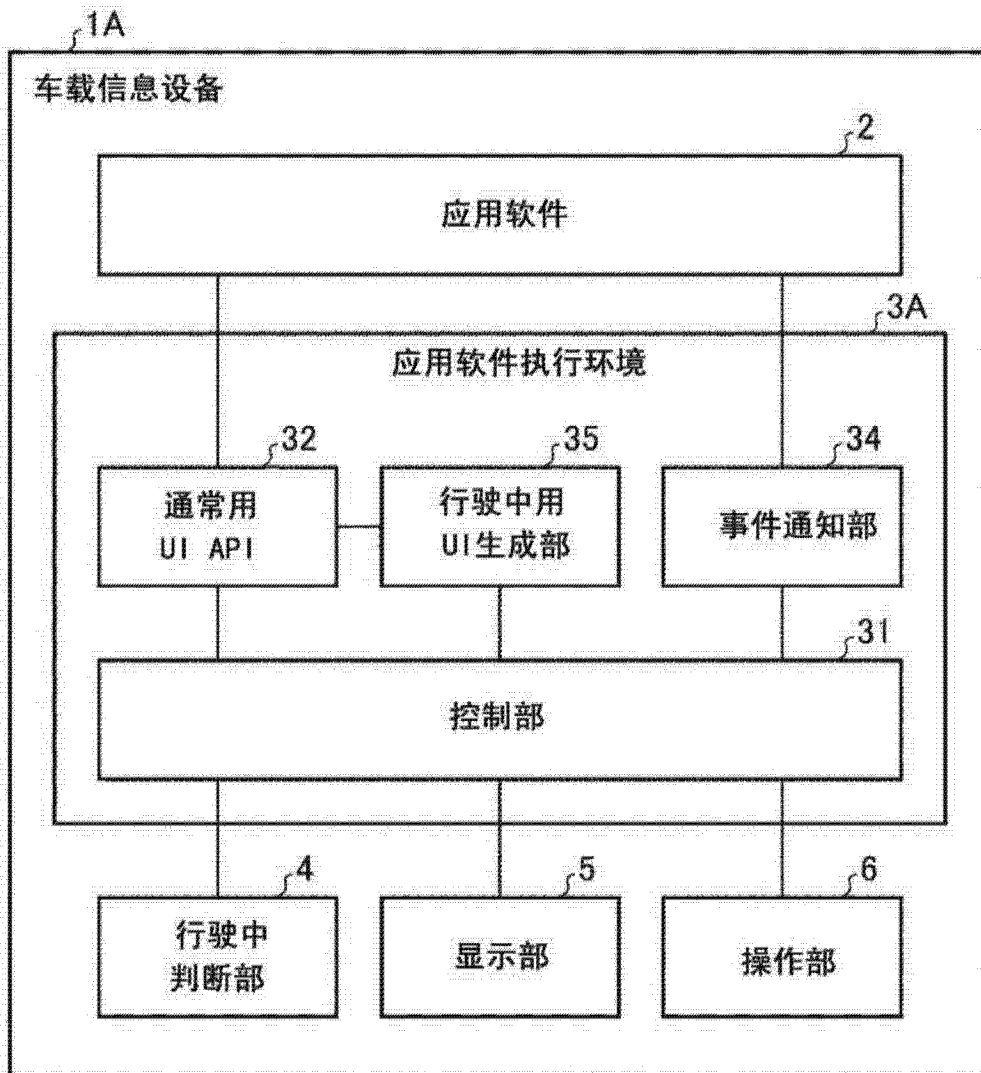


图 12

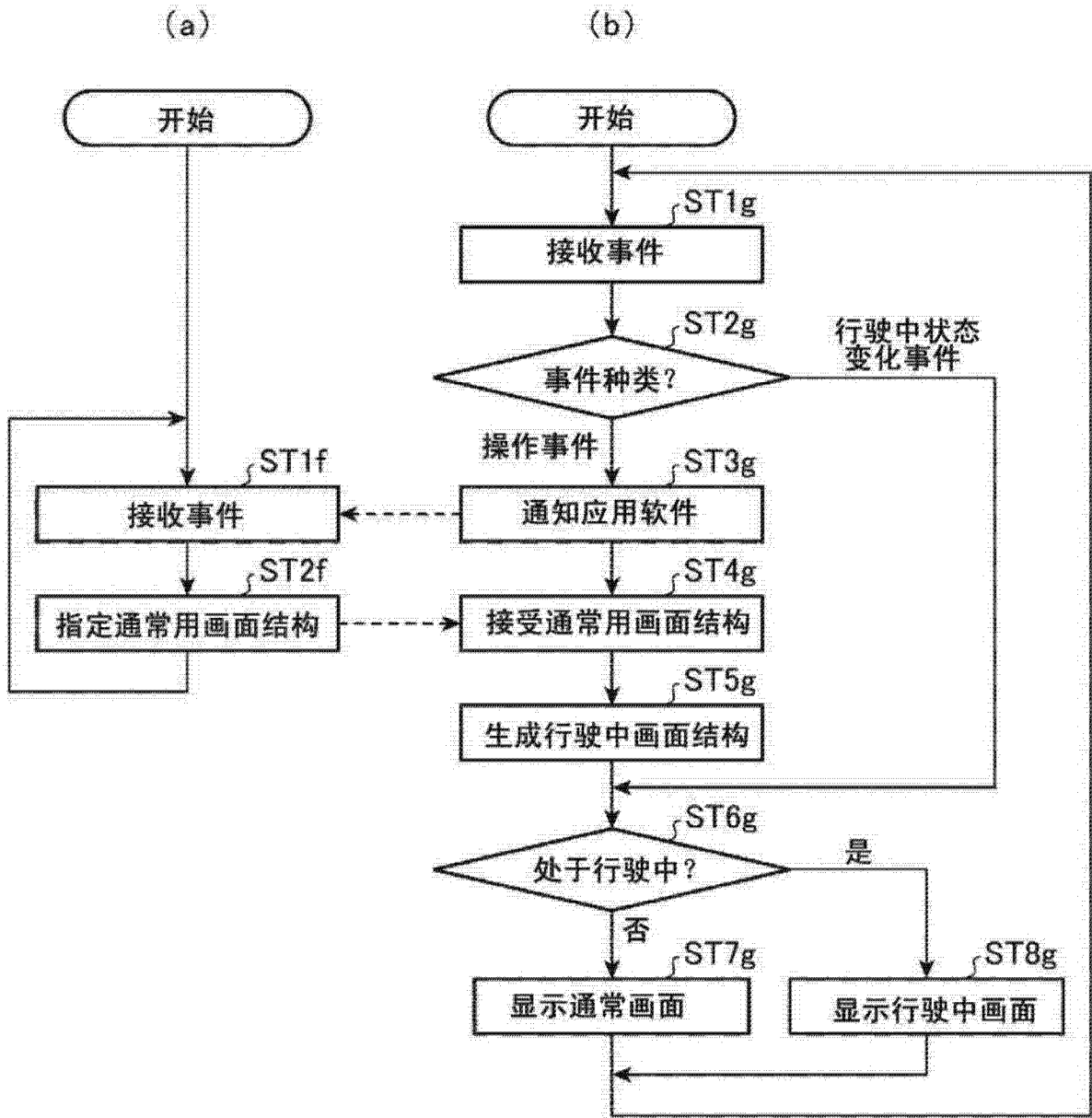


图 13

```

<!DOCTYPE running-ui>
<running-ui type="template-A">
<text id="msg1">新闻: 标题 </text>
<text id="btn1">返回 </button>
<text id="btn2">语音朗读 </button>
</running-ui>
  
```

图 14

```
<!DOCTYPE html>
<html>
<style type="text/css">
<!--
body, div, list {
padding: 2px;
margin: 2px;
border: solid 1px Black;
}
button {
width: 90px;
height: 2em;
}
.pagetitle {
background: LightGray;
}
.list {
width: 70%;
}
-->
</style>
<body>
<div class="pagetitle"> 新闻: 标题 </div>
<table>
<tr>
<td class="list">
<div class="heading" id="heading1">ABC获胜! </div>
<div class="heading" id="heading2">日元进一步走高 </div>
<div class="heading" id="heading3">DEF公司和GHI公司合作 </div>
</td>
<td>
</img>
</td>
</tr>
</table>
<button id="ret">返回 </button>
<button id="speech">语音朗读 </button>
<button id="prev-page">前一页 </button>
<button id="next-page">后一页
</body>
</html>
```

图 15



图 16

```
<!DOCTYPE running-ui>  
<running-ui type="template-C">  
<text id="msg1">新闻: 标题 </text>  
<text id="btn1">返回 </button>  
<text id="btn2">语音朗读 </button>  
<img id="img1">animation-fixed.gif</img>  
</running-ui>
```

图 17



图 18



```
<!DOCTYPE html>
<html>
<style type="text/css">
<!--
body, div, list {
padding: 2px;
margin: 2px;
border: solid 1px Black;
}
button {
width: 90px;
height: 2em;
}
.pagetitle {
background: LightGray;
}
-->
</style>
<body>
<running-ui type="template-A">
<div class="pagetitle" running-param="msg1">新闻: 标题 </div>
<div class="list">
<div class="heading" id="heading1">ABC获胜! </div>
<div class="heading" id="heading2">日元进一步走高</div>
<div class="heading" id="heading3">DEF公司和GHI公司合作 </div>
</div>
<button>返回 </button>
<button running-param="btn2">语音朗读 </button>
<button>前一页 </button>
<button>后一页 </button>
</body>
</html>
```

图 19

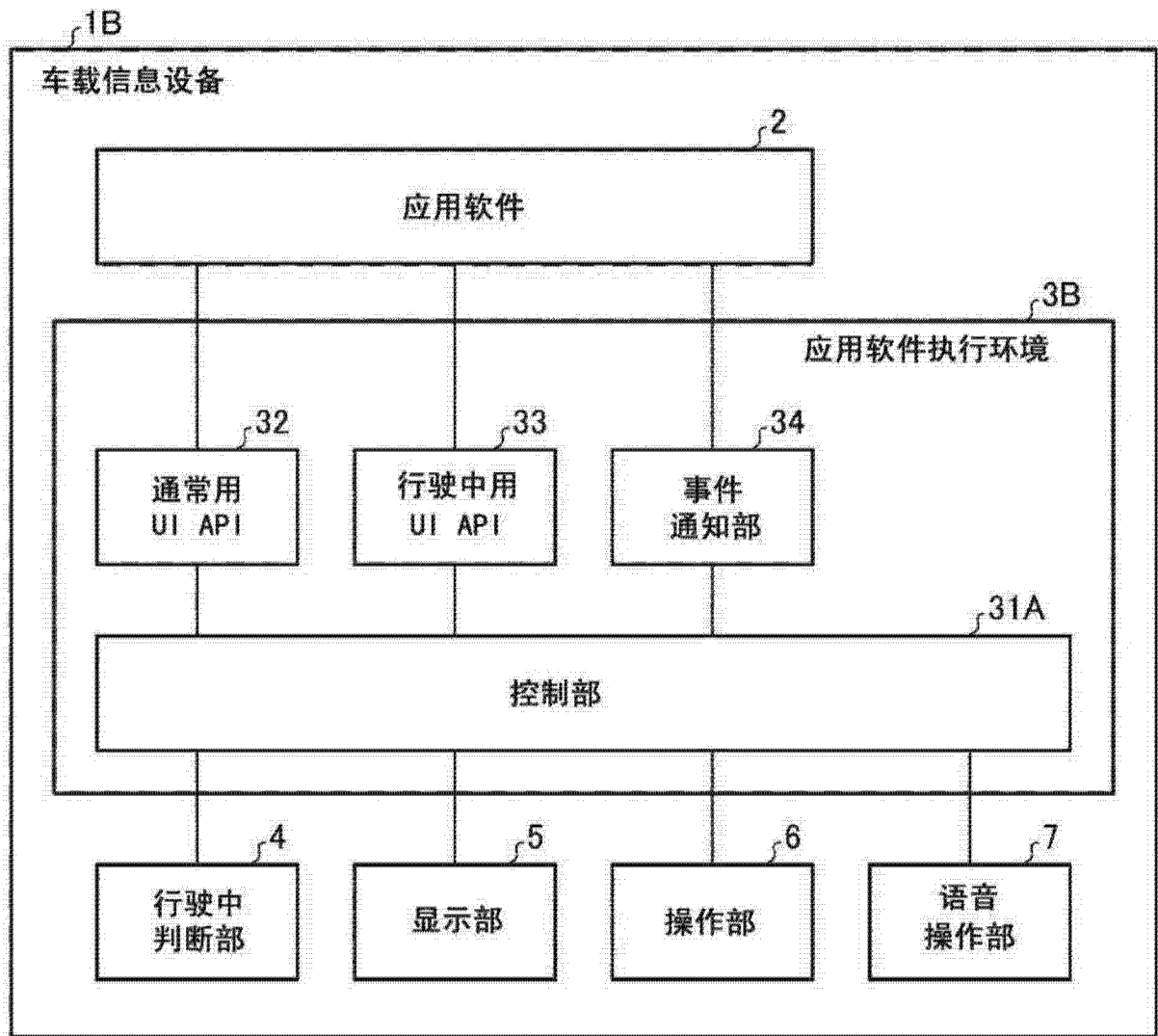


图 20

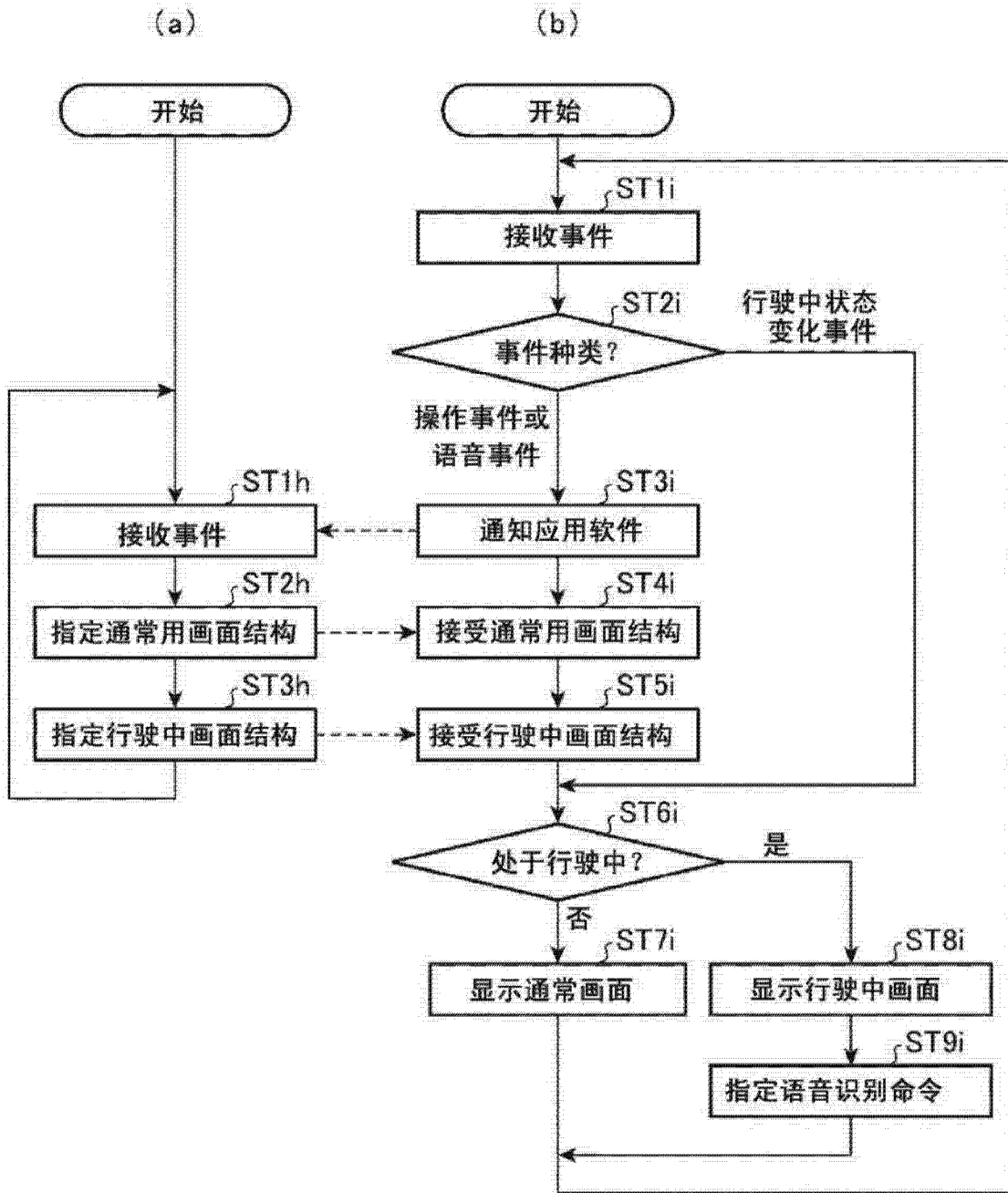


图 21

```
<!DOCTYPE running-ui>  
<running-ui type="template-A">  
<text id="msg1">新闻：标题 </text>  
<text id="btn2">语音朗读 </text>  
<speech> 返回 </speech>  
<speech> 语音朗读 </speech>  
</running-ui>
```

图 22