

虚拟机器人的控制方法、装置、设备、存储介质及程序产品

本申请要求在2022年05月16日提交中国专利局、申请号为202210529988.5的中国专利申请的优先权，该申请的全部内容通过引用结合在本申请中。

技术领域

本申请涉及计算机处理的技术领域，例如涉及一种虚拟机器人的控制方法、装置、设备、存储介质及程序产品。

背景技术

在游戏对战以及压力测试等业务场景中通常使用虚拟机器人仿真用户的行为，为此，要求技术人员针对特定的业务场景开发一套虚拟机器人的代码，并对这个虚拟机器人的代码进行维护，对虚拟机器人的行为进行拓展。

例如，针对直播的压力测试，可构建一个房间维度的虚拟机器人，该虚拟机器人定时触发任务，直播停止时清除该虚拟机器人。

这种虚拟机器人适用的业务场景单一，无论是开发还是维护均会涉及虚拟机器人的整套代码，均产生较大的工作量，导致成本较高。

发明内容

本申请提供了一种虚拟机器人的控制方法、装置、设备、存储介质及程序产品，以解决开发以及维护虚拟机器人的成本较高的问题。

根据本申请的一方面，提供了一种虚拟机器人的控制方法，包括：

启动多个使用有限状态自动机构建的虚拟机器人，所述虚拟机器人用于多次循环模拟用户操作应用程序，所述有限状态自动机包括多个状态、在所述多个状态下发生的多个事件、以及由所述多个事件触发的动作；

在每次循环中，确定所述虚拟机器人在所述应用程序中所处的视图，将所述视图作为所述状态；

在每次循环中，模拟所述用户在所述视图中选定服务，将选定的服务作为所述事件；

在每次循环中，驱动所述虚拟机器人在所述视图中执行用于触发所述服务的用户操作，将所述用户操作作为所述动作。

根据本申请的另一方面，提供了一种虚拟机器人的控制装置，包括：

虚拟机器人启动模块，设置为启动多个使用有限状态自动机构建的虚拟机器人，所述虚拟机器人用于多次循环模拟用户操作应用程序，所述有限状态自动机包括多个状态、在所述多个状态下发生的多个事件、由所述多个事件触发的动作；

状态确定模块，设置为在每次循环中，确定所述虚拟机器人在所述应用程序中所处的视图，将所述视图作为所述状态；

事件选定模块，设置为在每次循环中，模拟所述用户在所述视图中选定服务，将选定的服务作为所述事件；

动作执行模块，设置为在每次循环中，驱动所述虚拟机器人在所述视图中执行用于触发所述服务的用户操作，将所述用户操作作为所述动作。

根据本申请的另一方面，提供了一种虚拟机器人的控制设备，所述虚拟机器人的控制设备包括：

至少一个处理器；以及

与所述至少一个处理器通信连接的存储器；其中，

所述存储器存储有可被所述至少一个处理器执行的计算机程序，所述计算机程序被所述至少一个处理器执行，以使所述至少一个处理器能够执行本申请任一实施例所述的虚拟机器人的控制方法。

根据本申请的另一方面，提供了一种计算机可读存储介质，所述计算机可读存储介质存储有计算机程序，所述计算机程序用于使处理器执行时实现本申请任一实施例所述的虚拟机器人的控制方法。

根据本申请的另一方面，提供了一种计算机程序产品，包括计算机程序，所述计算机程序在被处理器执行时实现本申请任一实施例所述的虚拟机器人的控制方法。

附图说明

图1是根据本申请实施例一提供的一种虚拟机器人的控制方法的流程图；

图2是根据本申请实施例一提供的一种虚拟机器人运行示意图；

图3是根据本申请实施例一提供的一种视图的示例图；

图4是根据本申请实施例一提供的一种用户操作的示例图；

图5是根据本申请实施例二提供的一种虚拟机器人的控制方法的流程图；

图6是根据本申请实施例三提供的一种虚拟机器人的控制装置的结构示意

图；

图 7 是实现本申请实施例的虚拟机器人的控制方法的虚拟机器人的控制设备的结构示意图。

具体实施方式

下面将结合本申请实施例中的附图，对本申请实施例中的技术方案进行描述，所描述的实施例仅仅是本申请一部分的实施例，而不是全部的实施例。

本申请的说明书和权利要求书及上述附图中的术语“第一”、“第二”等是用于区别类似的对象，而不必用于描述特定的顺序或先后次序。这样使用的数据在适当情况下可以互换，以便这里描述的本申请的实施例能够以除了在这里图示或描述的那些以外的顺序实施。此外，术语“包括”和“具有”以及他们的任何变形，意图在于覆盖不排他的包含，例如，包含了一系列步骤或单元的过程、方法、系统、产品或设备不必限于列出的那些步骤或单元，而是可包括没有列出的或对于这些过程、方法、产品或设备固有的其它步骤或单元。

实施例一

图 1 为本申请实施例一提供的一种虚拟机器人的控制方法的流程图，本实施例可适用于基于有限状态自动机构建仿真虚拟机器人的情况，该方法可以由虚拟机器人的控制装置来执行，该虚拟机器人的控制装置可以采用硬件和/或软件的形式实现，该虚拟机器人的控制装置可配置于虚拟机器人的控制设备中。如图 1 所示，该方法包括以下步骤。

步骤 101、启动多个使用有限状态自动机构建的虚拟机器人。

在本实施例中，可以应用有限状态自动机（Finite State Machine, FSM）构建虚拟机器人，这些虚拟机器人属于计算机程序，可用于多次循环模拟用户去操作应用程序。

有限状态自动机又称状态机，是现实事物运行规则抽象而成的一个数学模型。

在部分情况中，有限状态自动机包括如下概念：

1、多个状态（State），可以指事物表现出来的形态，被描述的事物的状态数量是有限个，一般情况下，一个状态机至少要包含两个状态。

2、在状态下发生的多个事件（Event），可以指执行一个操作的触发条件或者口令。

3、由事件触发的动作（Action），可以指事件发生以后执行的操作，在编程的时候，一个动作一般就对应一个函数。

虚拟机器人模拟用户去操作应用程序，就是对用户仿真，参照用户的行为习惯对应用程序进行操作。

本实施例提供虚拟机器人运行的框架，该框架可适配不同的业务场景，例如，在游戏（应用程序）的练习模式中使用虚拟机器人模拟不同的用户与真实的用户进行对战，实现不同等级（如简单、中等、困难等）的对战联系，在压力测试中使用虚拟机器人模拟用户高频地对应用程序进行操作，测试应用程序可能存在的漏洞（Bug），在直播间（应用程序）中模拟观众用户观看直播、与主播用户互动，以仿真真实的观看环境，监控服务端与客户端的多种性能，为功能优化提供参考，等等。

在不同的业务场景中，可在该框架下添加与该业务场景适配的应用程序，例如，即时通讯工具、短视频应用、直播应用、新闻应用以及游戏，等等。

在不同的业务场景中，可在该框架下添加与该业务场景适配的用户信息，以对用户仿真，例如，用户标识（Identifier，ID）、用户的昵称、用户的头像、用户的等级以及用户所处的地区，等等。

这些用户信息可以在启动虚拟机器人时通过随机、带约束条件（如用户的年龄不允许低于一个数值）等方式配置，也可以在虚拟机器人在运行过程中，满足一些条件（如进入直播间等）的情况下通过随机、带约束条件等方式更换，等等，即，每个虚拟机器人既可以模拟一个用户的操作，也可以模拟多个用户的操作，本实施例对此不加以限制。

不同的应用程序具有不同的功能，相应地，虚拟机器人模拟用户的操作也有所不同，这些操作一般是与业务场景的需求相关，并不一定是应用程序可支持的所有操作，而是应用程序可支持的部分操作，例如，针对直播应用，直播应用的需求为为优化直播间的功能提供参考，那么，虚拟机器人模拟用户的操作可以包括进入主页、进入直播间、在直播间发弹幕信息、在直播间点赞、在直播间向主播用户赠送虚拟物品，虚拟机器人并不模拟用户的操作可以包括分享直播间、搜索关键词，等等。

如图2所示，在业务场景中，可以部署多个相互独立的虚拟机器人，如虚拟机器人1、虚拟机器人2，……，虚拟机器人n，形成虚拟机器人集群，所谓相互独立，可以指使用独立的虚拟机、独立的线程等方式实现虚拟机器人，虚拟机器人之间并不存在关联性，虚拟机器人的运行、仿真均是独立的，相互之间并不影响。

示例性地，在每个虚拟机器人的实例中对虚拟机器人进行哈希分片，即，每个虚拟机器人的实例处理部分虚拟机器人的动作，提高虚拟机器人的扩容性。

这些虚拟机器人的运行策略（如每个虚拟机器人运行的时间、一个时间段内虚拟机器人运行的数量等）可以预先针对业务场景配置，从而按照该运行策略启动相应的虚拟机器人，例如，在傍晚时间使用更多的虚拟机器人模拟数量更多的用户进入直播间，在凌晨时间使用更少的虚拟机器人模拟数量更少的用户进入直播间，等等。

此外，在容灾以及故障重启等情况下，也会重新启动虚拟机器人，恢复原有的业务场景。

针对每个虚拟机器人，可设置任务队列，该任务队列用于存储任务，按照业务场景的业务需求设置分发任务的策略（如分发的时间周期等），按照该策略将任务分发至虚拟机器人，虚拟机器人执行任务，模拟用户去操作应用程序，虚拟机器人模拟一次用户的操作会遍历一遍有限状态自动机，这个过程可称之为一次 tick 循环，从用户的角度看，真实的用户一般会在应用程序执行多种操作，从应用程序的角度看，在同一时刻，用户之间共同访问的空间（如直播间、游戏中的场景等）会存在多个用户的操作，那么，为仿真这些情景，虚拟机器人会多次 tick 循环模拟同一用户或不同的用户去操作应用程序。

步骤 102、在每次循环中，确定虚拟机器人在应用程序中所处的视图，将该视图作为状态。

如图 2 所示，有限状态自动机可提供多个状态（State），如状态 0-状态 6，在每一次 tick 循环中，可以识别虚拟机器人在应用程序中当前所处的视图（View），按照有限状态自动机的机制，该视图（View）为有限状态自动机中的状态（State）。

虚拟机器人并不一定会加载应用程序的用户界面（User Interface, UI），而是预先针对每个视图（View）设置标志位，使用标志位表示在应用程序中当前所处的视图（View），从而避免加载 UI 的操作，降低虚拟机器人的功耗。

视图（View）是安卓（Android）等操作系统中空间的基类，不管是简单的文本视图（TextView），按钮（Button）还是复杂的线性布局（LinearLayout）和列表视图（ListView），它们的共同基类都是 View。

View 是一种界面层的控件的一种抽象，它代表了一个控件，View 的形式还有控件组（ViewGroup），即一组 View，ViewGroup 也继承了 View，这就意味着 View 可以是单个控件，也可以是由多个控件组成的一组控件。

对于压力测试等业务场景中，考虑到业务场景的业务需求，可以加载应用

程序的 UI，从而查询当前 UI 中焦点所处的视图（View），本实施例对此不加限制。

针对不同的应用程序，其所提供的视图（View）也有所不同，在虚拟机器人模拟用户操作应用程序的过程中，根据业务需求可以将应用程序的全部视图（View）设置为有限状态自动机中的状态，也可以将应用程序的部分视图（View）设置为有限状态自动机中的状态，而忽略应用程序的部分视图（View）。

以直播应用为例，如图 3 所示，在一个直播应用中，可提供首页、个人页面、直播间游戏、私信、抽奖等视图（View），若虚拟机器人处于首页，则可以模拟用户选择个人页面、直播间游戏、私信、抽奖，也可以模拟用户停留在首页。

本实施例以视图（View）为有限状态自动机中的状态，可以提高用户在应用程序执行的操作的覆盖率，提高虚拟机器人的仿真的全面性。

针对虚拟机器人可以预先配置数据库，该数据库可用于按照循环从时间顺序为每个虚拟机器人（以 ID 等标识表示）存储该虚拟机器人在有限状态自动机中所处的状态，即，为每个虚拟机器人存储该虚拟机器人在应用程序中所处的视图（View）。

示例性地，可以以诸如 robot_123456789_status 的哈希数据结构来保存标识为 123456789 的虚拟机器人的状态。

在启动虚拟机器人进行首次循环中，可以依据以 ID 等标识从预置的数据库中读取在上一次循环中为虚拟机器人存储的、有限状态自动机中的状态。

若读取成功，即，虚拟机器人在先具有特定的状态，则将该状态配置给虚拟机器人，以恢复应用程序中的视图（View），使得每个虚拟机器人可以恢复自身的状态。

若读取失败，即，虚拟机器人在先没有特定的状态，则可以确定虚拟机器人在应用程序中处于默认的视图（View），如应用程序的首页等，将该默认的视图作为有限状态自动机中的状态。

在一些业务场景中，如果出现容灾、故障重启等情况，重新启动虚拟机器人，这些虚拟机器人在原有的业务场景中已经在持续模拟用户操作（即用户可执行的操作），分布在不同的视图（View）中，容灾、故障重启等情况会中断虚拟机器人的状态，重启之后恢复虚拟机器人的状态可以避免虚拟机器人集中出现在一些默认的视图（View）中，使得业务场景更加自然，真实度更高。

对于非首次启动，虚拟机器人可去内存等区域读取上一次 tick 循环时所处的状态。

步骤 103、在每次循环中，模拟用户为虚拟机器人在视图中选定服务，将选定的服务作为事件。

在不同的应用程序中，不同的视图（View）可以提供不同的服务，例如，在直播间的视图（View）中，可提供观看直播的服务、发布弹幕的服务、给主播用户赠送虚拟物品的服务、点赞的服务，等等。

如图 2 所示，在每次 tick 循环中，可以模拟用户的行为习惯，为虚拟机器人在当前的视图（View）中选定其中一个服务，用户为得到这个服务会触发一个用户操作，按照有限状态自动机的机制，该服务可为有限状态自动机中在状态下发生的事件（Event）（如事件 1-事件 n）。

在本申请的一个实施例中，步骤 103 可以包括如下步骤。

步骤 1031、在每次循环中，加载为视图配置的第一文件。

步骤 1032、在第一文件中读取视图提供的服务、以及筛选规则。

本实施例提供虚拟机器人运行的框架，针对不同的业务场景，技术人员可以遵循该框架的规范，预先对应用程序中的多个视图（View）配置第一文件，第一文件的格式可以包括可扩展标记语言（Extensible Markup Language, XML）、动态链接库（Dynamic Link Library, DLL）等，第一文件可以以配置文件、插件、脚本等方式实现，该第一文件中记录每个视图（View）可提供的服务、以及筛选规则，该筛选规则即为，用于筛选每个服务的、且符合用户的行为习惯的规范。

在每次 tick 循环中，可加载为当前视图（View）配置的第一文件至内存运行，从而在第一文件中读取当前视图（View）提供的多个服务及相应的筛选规则。

步骤 1033、模拟用户按照筛选规则选择其中一个服务，将选择的服务作为有限状态自动机中在状态下发生的事件。

对筛选规则进行解析，执行筛选规则记录的规范，从而模拟用户从所有的服务中选择其中一个服务，作为有限状态自动机中在状态下发生的事件。

在一个示例中，筛选规则中记录有每个服务与概率之间的映射关系，表示按照用户的行为习惯选择每个服务的概率。

在本示例中，从筛选规则中读取为用户选择每个服务的概率。

按照概率为虚拟机器人选择其中一个服务，作为有限状态自动机中在状态下发生的事件。

服务包括闲置（Idle），所谓闲置，可以是指用户处于观看和/或收听视图

(View) 中的信息 (如直播画面、新闻、小说等)、用户不在视图 (View) 前等情况, 并不对视图 (View) 执行有效的用户操作。

在本示例中, 闲置的概率大于除闲置外任一服务的概率, 即, 大概率会选择闲置这个服务, 去模拟实现用户不会频繁在应用程序中的视图 (View) 执行有效的用户操作的情况。

以直播应用为例, 如图 4 所示, 若虚拟机器人处于直播间 (状态 (State)), 提供的服务包括闲置、退房、发布弹幕信息、赠送虚拟物品与点赞等等, 用户选择闲置的概率 (如 90%) 大于用户选择退房的概率 (如 4%)、发布弹幕信息的概率 (如 5%)、赠送虚拟物品的概率 (如 0.4%) 或点赞的概率 (如 0.6%)。

在本申请的另一个实施例中, 步骤 103 可以包括如下步骤。

步骤 1034、在视图中检测到错误。

步骤 1035、响应于错误, 模拟用户在视图中选定闲置, 将闲置作为有限状态自动机中在状态下发生的事件。

在本实施例中, 视图 (View) 提供的服务包括闲置。

虚拟机器人的状态 (即视图 (View)) 在数据不一致使得当前状态无法执行一个动作等情况中会出现错误, 例如, 在直播间中主播用户停止直播, 导致虚拟机器人无法按照直播的逻辑执行循环, 等等, 此时, 会忽略原有设置的筛选规则, 强制选定闲置 (Idle) 为有限状态自动机中在状态下发生的事件, 等待下一次 tick 循环重新触发, 通过闲置来避免触发错误的用户操作, 保证模拟用户操作的真实性。

步骤 104、在每次循环中, 驱动虚拟机器人在视图中执行用于触发服务的用户操作, 将用户操作作为动作。

如图 2 所示, 在每一次 tick 循环中, 驱动虚拟机器人在视图 (View) 中执行用于触发服务的用户操作, 按照有限状态自动机的机制, 该用户操作为有限状态自动机中由事件 (Event) 触发的动作 (Action) (如动作 1-动作 n), 此时, 可返回任务队列, 等待下一次任务, 等待下一次 tick 循环。

在本申请的一个实施例中, 步骤 104 可以包括如下步骤。

步骤 1041、在每次循环中, 加载为视图配置的第二文件。

步骤 1042、在第二文件中读取为服务配置的模拟规则。

本实施例提供虚拟机器人运行的框架, 针对不同的业务场景, 技术人员可以遵循该框架的规范, 预先对应用程序中的多个视图 (View) 配置第二文件, 第二文件的格式可以包括 XML、DLL 等, 第二文件可以以配置文件、插件、脚

本等方式实现，该第二文件中记录多个服务的模拟规则，即，用于触发实现多个服务的用户操作的规范。

第一文件与第二文件可以为同一份文件，也可以为不同文件，本实施例对此不加以限制。

在每次 tick 循环中，可加载为当前视图（View）配置的第二文件至内存运行，从而在第二文件中读取选定的服务配置的模拟规则。

步骤 1043、驱动虚拟机器人在视图中按照模拟规则执行用户操作，将用户操作作为有限状态自动机中由事件触发的动作。

对模拟规则进行解析，执行模拟规则记录的规范，从而驱动虚拟机器人在视图（View）中按照模拟规则执行相应的用户操作，作为有限状态自动机中由事件（Event）触发的动作（Action）。

在实施例中，对于应用程序本地执行的用户操作，可以直接在应用程序本地执行。

对于与服务器交互的用户操作，应用程序提供了多个实现用户操作的应用程序接口（Application Programming Interface, API）的接口规范，一般情况下，用户在应用程序的 UI 上执行用户操作，则应用程序会按照在 UI 上被触发的元素封装相应的用户请求，并将用户请求发送至服务器，而在本实施例中，这些接口规范记录在模拟规则中，那么，可以在模拟规则中读取执行用户操作的接口规范。

虚拟机器人并不一定加载应用程序的 UI，而是直接按照接口规范生成用于表示在视图（View）中执行用户操作的用户请求，驱动虚拟机器人发送用户请求至服务器，作为有限状态自动机中由事件（Event）触发的动作（Action）。

例如，如图 4 所示，若选定服务为退房、点赞，则可以按照接口规范封装相应的用户请求，从而向服务器请求退出直播间、对该直播间点赞。

本实施例直接封装用户请求实现用户操作，避免加载 UI 的操作，从而降低虚拟机器人的功耗。

对于压力测试等业务场景中，考虑到业务场景的业务需求，可以加载应用程序的 UI，从而在当前 UI 中模拟用户操作，如点击、长按等，本实施例对此不加以限制。

在一种执行用户操作的方式中，可以采集虚拟机器人在模拟用户操作应用程序的过程中记录的上下文信息，该上下文信息用于辅助触发用户操作，并将这些上下文信息写入数据库中。

由于上下文的读写较为频繁，属于多读多写，因此，数据库可选为内存键值对（key/value，KV）数据库，如 Redis、MemCache 等。

针对不同的业务场景，应用程序中可执行的用户操作有所不同，所存储的上下文信息也有所不同，本实施例对此不加以限制。

以直播为例，存储的上下文信息如下：

key	value
虚拟机器人的标识	123123
今天发言数	34
今天送礼数	2
最近发言	[“aaa” , “bbb” , “ccc”]
发言决策网络	神经网络参数
包裹内容	[“111” , “222” , “333”]

在本方式中，如图 2 所示，可从预置的数据库中获取虚拟机器人在模拟用户操作应用程序的过程中记录的上下文信息。

按照模拟规则从所有的上下文信息中读取当前服务所依赖的第一环境信息，所谓依赖，可以指实现该服务的用户操作与第一环境信息之间存在使用的关系，第一环境信息对实现该服务的用户操作存在影响，反之，为触发实现该服务的用户操作需要第一环境信息的协助。

按照模拟规则确定服务依赖的组件（如第三方的发言决策网络，以 ID、路径等标识表示），将第一环境信息输入该组件，从而调用该组件生成与第一环境信息适配的目标信息，例如，文本信息、表情，等等，驱动虚拟机器人在视图（View）中发布目标信息，作为有限状态自动机中由事件（Event）触发的动作（Action）。

在一些业务场景中，用户会在应用程序发布信息，例如，如图 4 所示，在直播间中发布弹幕信息，在新闻下发布评论，等等，用户大多是针对特定的业务对象（如直播节目、新闻等）发布的信息，这些信息是体现用户对业务对象的看法，并非无意义的信息，因此，组件可以直接通过计算机视觉的技术分析业务对象（上下文信息）本身的语义，从而生成通过自然语言处理（Natural Language Processing, NLP）用于表达该语义的目标信息，此外，由于视频数据等格式的业务对象较难快速分析语义，而其他用户针对业务对象发布的信息（即

上下文信息，如其他用户在直播间中已发布的弹幕信息）体现了其他用户对该业务对象的看法，因此，组件可以通过 NLP 分析其他用户针对业务对象发布的信息的语义，从而生成与该语义相同或相似的目标信息（如弹幕信息），从而发布该目标信息（如弹幕信息）。

在另一种执行用户操作的方式中，如图 2 所示，可从预置的数据库中获取虚拟机器人在模拟用户操作应用程序的过程中记录的上下文信息，按照模拟规则从上下文信息中读取限制服务的第二环境信息。

所谓限制，一方面是指一个真实的用户，在应用程序中实际执行用户操作所受到自然人的能力、兴趣等生理限制，大概率不会出现频繁执行用户操作，漫无目的执行用户操作等情况，超过这些限制会使得用户的真实性降低，例如，用户在直播间中多是间断性发布弹幕信息，较小概率持续发布多个弹幕信息，青年阶段的用户较小概率持续观看婴幼儿阶段的动画、并对该婴幼儿阶段的动画发布评论，等等。

所谓限制，另一方面指应用程序所提供的用户操作是受到服务端的业务限制，大概率不会出现突破业务限制的情况，例如，在直播间禁言的情况下，任一观众用户均不能发布信息，又例如，如图 4 所示，在直播间限定一个虚拟物品的数量为 m 时，观众用户赠送虚拟物品的数量并不能大于限定的数量 m ，在并不大于数量 m 的情况下，观众用户可以赠送虚拟物品给主播用户，等等。

按照模拟规则确定限制该服务的条件，这些条件可能是来自于用户的生理限制、也可能来自于服务端的业务限制，例如，在直播间中赠送虚拟物品的数量，在直播间发布弹幕信息的时间间隔，在直播间发布弹幕信息的数量，等等。

将同一服务对应的第二环境信息与条件进行比较。

若第二环境信息未满足该条件，则允许驱动虚拟机器人在视图（View）中按照模拟规则执行用户操作，作为有限状态自动机中由事件（Event）触发的动作（Action）。

若第二环境信息满足该条件，则禁止驱动虚拟机器人在视图中按照模拟规则执行用户操作，作为有限状态自动机中由事件（Event）触发的动作（Action）。

在本申请的另一个实施例中，步骤 104 可以包括如下步骤。

步骤 1044、每次循环中，若服务为闲置，则确定用于触发服务的用户操作为空。

步骤 1045、响应于用户操作，驱动虚拟机器人在视图中等待，作为有限状态自动机中由事件触发的动作。

在每次 tick 循环中，可以检测服务的类型，如果服务的类型为闲置 (Idle)，则可以将用于触发服务的用户操作设置为空。

此时，驱动虚拟机器人在当前视图 (View) 中等待，暂时不执行有效的用户操作，按照有限状态自动机的机制，等待为有限状态自动机中由事件 (Event) 触发的动作 (Action)。

在本实施例中，启动多个使用有限状态自动机构建的虚拟机器人，虚拟机器人用于多次循环模拟用户操作应用程序，在每次循环中，确定虚拟机器人在应用程序中所处的视图，作为有限状态自动机中的状态，在每次循环中，模拟用户为虚拟机器人在视图中选定服务，作为有限状态自动机中在状态下发生的事件，在每次循环中，驱动虚拟机器人在视图中执行用于触发服务的用户操作，作为有限状态自动机中由事件触发的动作。

本实施例使用有限状态自动机构建虚拟机器人，提供了虚拟机器人在有限状态自动机的规范下运行的框架，方便将该框架集成到不同业务场景的模块中，并编写业务场景的逻辑，通用性强，扩展性强。

例如，如果一个系统整个接入了虚拟机器人，在压测工具、单元测试工具等第三方的工具提供接口给虚拟机器人的情况下，这些接口就可以集成到虚拟机器人的框架中，方便虚拟机器人挂钩 (hook) 第三方的工具，实现压测、单元测试等业务需求。

利用有限状态自动机设计的虚拟机器人的框架具有高维护性，业务人员不用关心有限状态自动机的底层框架，不用开发及维护时关心上层的状态以及动作，不仅使得业务代码可以按照状态来拆分，即，业务代码之间基本全部解耦，实现高耦合、低内聚，大大降低了开发及维护时的工作量，从而大大降低了成本，而且方便拓展虚拟机器人的行为，丰富虚拟机器人的行为，从而高自由度地模拟用户操作，提高虚拟机器人仿真用户的真实性。

虚拟机器人所保存的信息量少，对存储的压力很小，并且，虚拟机器人模拟真实的用户，动作并不频繁，主要在触发具体的动作时占用处理器等资源，扩容性强，在重启或容灾时，水平扩容即可。

实施例二

图 5 为本申请实施例二提供的一种虚拟机器人的控制方法的流程图，本实施例在上述实施例的基础上增加了存储上下文信息、对有限状态自动机中的状态进行变换的操作。如图 5 所示，该方法包括以下步骤。

步骤 501、启动多个使用有限状态自动机构建的虚拟机器人。

虚拟机器人用于多次循环模拟用户操作应用程序，有限状态自动机包括多个状态、在状态下发生的多个事件、由事件触发的动作。

步骤 502、在每次循环中，确定虚拟机器人在应用程序中所处的视图，作为状态。

步骤 503、在每次循环中，模拟用户在视图中选定服务，作为事件。

步骤 504、在每次循环中，驱动虚拟机器人在视图中执行用于触发服务的用户操作，作为动作。

步骤 505、若完成执行用户操作，则获取在视图中出现的上下文信息。

步骤 506、将上下文信息存储至预置的数据库中，并在数据库中建立虚拟机器人与上下文信息之间的关联关系。

如图 2 所示，若虚拟机器人执行完用户操作，则可以按照业务场景的业务需求，从应用程序的多个视图（View）中获取上下文信息，将上下文信息以键值对等形式记录。

以虚拟机器人（以 ID 等标识表示）作为索引，在数据库中建立虚拟机器人与上下文信息之间的关联关系，以便后续以虚拟机器人为条件在该数据库中查询相应的上下文信息，从而辅助执行后续的用户操作。

一般情况下，当虚拟机器人销毁时，可释放至少部分上下文信息，从而降低数据库中存储空间占用。

步骤 507、若完成执行用户操作，则确定虚拟机器人在应用程序中所处的视图。

步骤 508、按照执行用户操作之后的视图更新有限状态自动机中的状态，作为有限状态自动机中的变换。

在部分情况中，如图 2 所示，有限状态自动机包括如下概念：

1、状态（State）

2、事件（Event）

3、动作（Action）

4、表示状态变换的变换（Transition），变换指从一个状态变化为另一个状态。

在一些情况中，虚拟机器人执行用户操作之后，应用程序从一个视图（View）变换至另一个视图（View），此时，可以查询虚拟机器人在应用程序中所处的、新的视图（View），按照新的视图（View）更新有限状态自动机中的状态（State），

此时的状态 (State) 会存储至数据库中, 按照有限状态自动机的机制, 这个过程为有限状态自动机中的变换 (Transition), 保证有限状态自动机的正常运行, 从而保证虚拟机器人的正常运行。

例如, 如图 4 所示, 虚拟机器人在执行完退房对应的动作之后, 从直播间 (状态 (State)) 变换至首页 (状态 (State))。

在一些情况中, 例如当服务为闲置 (Idle) 时, 虚拟机器人执行用户操作之后, 应用程序保持当前的视图 (View) 不变, 此时, 维持有限状态自动机中的状态 (State), 并不会更新有限状态自动机中的状态 (State)。

例如, 如图 4 所示, 虚拟机器人在执行完闲置、发布弹幕信息、赠送虚拟物品或点赞对应的动作之后, 维持直播间 (状态 (State)) 不变。

实施例三

图 6 为本申请实施例三提供的一种虚拟机器人的控制装置的结构示意图。如图 6 所示, 该装置包括以下模块。

虚拟机器人启动模块 601, 设置为启动多个使用有限状态自动机构建的虚拟机器人, 所述虚拟机器人用于多次循环模拟用户操作应用程序, 所述有限状态自动机包括多个状态、在所述多个状态下发生的多个事件、以及由所述多个事件触发的动作;

状态确定模块 602, 设置为在每次循环中, 确定所述虚拟机器人在所述应用程序中所处的视图, 将所述视图作为所述状态;

事件选定模块 603, 设置为在每次循环中, 模拟所述用户在所述视图中选定服务, 将选定的服务作为所述事件;

动作执行模块 604, 设置为在每次循环中, 驱动所述虚拟机器人在所述视图中执行用于触发所述服务的用户操作, 将所述用户操作作为所述动作。

在本申请的一个实施例中, 所述状态确定模块 602 包括:

状态读取模块, 设置为在首次循环中, 从预置的数据库中读取在上一次循环中为所述虚拟机器人存储的、所述有限状态自动机中的状态;

状态恢复模块, 设置为若读取成功, 则将所述状态配置给所述虚拟机器人, 以恢复所述应用程序中的视图;

默认设置模块, 设置为若读取失败, 则确定所述虚拟机器人在所述应用程序中处于默认的视图, 将所述默认的视图作为所述状态。

在本申请的一个实施例中，所述事件选定模块 603 包括：

第一文件加载模块，设置为在每次循环中，加载为所述视图配置的第一文件；

筛选规则读取模块，设置为在所述第一文件中读取所述视图提供的服务、以及筛选规则；

规则筛选模块，设置为模拟所述用户按照所述筛选规则选择其中一个服务，将选择的服务作为所述事件。

在本申请的一个实施例中，所述规则筛选模块包括：

概率读取模块，设置为从所述筛选规则中读取为所述用户选择每个服务的概率；

概率选择模块，设置为按照所述概率为所述虚拟机器人选择其中一个服务，作为所述事件；

其中，所述服务包括闲置，所述闲置的概率大于除所述闲置外任一服务的概率。

在本申请的一个实施例中，所述事件选定模块 603 还设置为：

在所述视图中检测到错误，其中，所述视图提供的服务至少包括闲置；

响应于所述错误，模拟所述用户在所述视图中选定所述闲置，将所述闲置作为所述事件。

在本申请的一个实施例中，所述动作执行模块 604 包括：

第二文件加载模块，设置为在每次循环中，加载为所述视图配置的第二文件；

模拟规则读取模块，设置为在所述第二文件中读取为所述服务配置的模拟规则；

模拟规则执行模块，设置为驱动所述虚拟机器人在所述视图中按照所述模拟规则执行用户操作，作为所述动作。

在本申请的一个实施例中，所述模拟规则执行模块是设置为：

在所述模拟规则中读取执行所述用户操作的接口规范；

按照所述接口规范生成用于表示在所述视图中执行所述用户操作的用户请求；

驱动所述虚拟机器人发送所述用户请求至服务器，作为所述动作。

在本申请的一个实施例中，所述模拟规则执行模块是设置为：

从预置的数据库中获取所述虚拟机器人在模拟所述用户操作所述应用程序的过程中记录的上下文信息；

按照所述模拟规则从所述上下文信息中读取所述服务依赖的第一环境信息；

按照所述模拟规则确定所述服务依赖的组件；

调用所述组件生成与所述第一环境信息适配的目标信息；

驱动所述虚拟机器人在所述视图中发布所述目标信息，作为所述动作。

在本申请的一个实施例中，所述模拟规则执行模块是设置为：

从预置的数据库中获取所述虚拟机器人在模拟所述用户操作所述应用程序的过程中记录的上下文信息；

按照所述模拟规则从所述上下文信息中读取限制所述服务的第二环境信息；

按照所述模拟规则确定限制所述服务的条件；

在所述第二环境信息未满足所述条件的情况下，允许驱动所述虚拟机器人在所述视图中按照所述模拟规则执行所述用户操作，作为所述动作；

在所述第二环境信息满足所述条件的情况下，禁止驱动所述虚拟机器人在所述视图中按照所述模拟规则执行用户操作，作为所述动作。

在本申请的另一个实施例中，所述动作执行模块 604 包括：

空操作确定模块，设置为每次循环中，若所述服务为闲置，则确定用于触发所述服务的用户操作为空；

等待模块，设置为响应于所述用户操作，驱动所述虚拟机器人在所述视图中等待，作为所述动作。

在本申请的一个实施例中，还包括：

上下文信息获取模块，设置为若完成执行所述用户操作，则获取在所述视图中出现的上下文信息；

上下文信息存储模块，设置为将所述上下文信息存储至预置的数据库中，并在所述数据库中建立所述虚拟机器人与所述上下文信息之间的关联关系。

在本申请的一个实施例中，所述有限自动状态机还包括表示所述状态变换的变换，该装置还包括：

新视图确定模块，设置为若完成执行所述用户操作，则确定所述虚拟机器人在所述应用程序中所处的视图；

状态变换模块，设置为按照执行所述用户操作之后的视图更新所述有限状态自动机中的状态，作为所述变换。

本申请实施例所提供的虚拟机器人的控制装置可执行本申请任意实施例所提供的虚拟机器人的控制方法，具备执行虚拟机器人的控制方法相应的功能模块和效果。

实施例四

图 7 示出了可以用来实施本申请的实施例的虚拟机器人的控制设备 10 的结构示意图。

如图 7 所示，虚拟机器人的控制设备 10 包括至少一个处理器 11，以及与至少一个处理器 11 通信连接的存储器，如只读存储器(Read-Only Memory, ROM) 12、随机访问存储器(Random Access Memory, RAM) 13 等，其中，存储器存储有可被至少一个处理器执行的计算机程序，处理器 11 可以根据存储在 ROM 12 中的计算机程序或者从存储单元 18 加载到 RAM 13 中的计算机程序，来执行多种适当的动作和处理。在 RAM 13 中，还可存储虚拟机器人的控制设备 10 操作所需的多种程序和数据。处理器 11、ROM 12 以及 RAM 13 通过总线 14 彼此相连。输入/输出(Input/Output, I/O) 接口 15 也连接至总线 14。

虚拟机器人的控制设备 10 中的多个部件连接至 I/O 接口 15，包括：输入单元 16，例如键盘、鼠标等；输出单元 17，例如多种类型的显示器、扬声器等；存储单元 18，例如磁盘、光盘等；以及通信单元 19，例如网卡、调制解调器、无线通信收发机等。通信单元 19 允许虚拟机器人的控制设备 10 通过诸如因特网的计算机网络和/或多种电信网络与其他设备交换信息/数据。

处理器 11 可以是多种具有处理和计算能力的通用和/或专用处理组件。处理器 11 的一些示例包括中央处理单元(Central Processing Unit, CPU)、图形处理单元(Graphics Processing Unit, GPU)、多种专用的人工智能(Artificial Intelligence, AI) 计算芯片、多种运行机器学习模型算法的处理器、数字信号处理器(Digital Signal Processor, DSP)、以及任何适当的处理器、控制器、微控制器等。处理器 11 执行上文所描述的多个方法和处理，例如虚拟机器人的控制方法。

在一些实施例中，虚拟机器人的控制方法可被实现为计算机程序，其被

有形地包含于计算机可读存储介质，例如存储单元 18。在一些实施例中，计算机程序的部分或者全部可以经由 ROM 12 和/或通信单元 19 而被载入和/或安装到虚拟机器人的控制设备 10 上。当计算机程序加载到 RAM 13 并由处理器 11 执行时，可以执行上文描述的虚拟机器人的控制方法的一个或多个步骤。备选地，在其他实施例中，处理器 11 可以通过其他任何适当的方式（例如，借助于固件）而被配置为执行虚拟机器人的控制方法。

实施例五

本申请实施例还提供了一种计算机程序产品，包括计算机程序，该计算机程序在被处理器执行时实现如本申请任一实施例所提供的虚拟机器人的控制方法。

计算机程序产品在实现的过程中，可以以一种或多种程序设计语言或其组合来编写用于执行本申请操作的计算机程序代码，程序设计语言包括面向对象的程序设计语言，诸如 Java、Smalltalk、C++，还包括常规的过程式程序设计语言，诸如“C”语言或类似的程序设计语言。程序代码可以完全地在用户计算机上执行、部分地在用户计算机上执行、作为一个独立的软件包执行、部分在用户计算机上部分在远程计算机上执行、或者完全在远程计算机或服务器上执行。在涉及远程计算机的情形中，远程计算机可以通过任意种类的网络——包括局域网（Local Area Network, LAN）或广域网（Wide Area Network, WAN）——连接到用户计算机，或者，可以连接到外部计算机（例如利用因特网服务提供商来通过因特网连接）。

本申请实施例还提供一种包含计算机可执行指令的存储介质，所述计算机可执行指令在由计算机处理器执行时用于执行本申请实施例提供的虚拟机器人的控制方法。存储介质可以是非暂态（non-transitory）存储介质。

可以使用上面所示的多种形式的流程，重新排序、增加或删除步骤。例如，本申请中记载的多个步骤可以并行地执行也可以顺序地执行也可以不同的次序执行，只要能够实现本申请的技术方案所期望的结果，本文在此不进行限制。

权 利 要 求 书

1、一种虚拟机器人的控制方法，包括：

启动多个使用有限状态自动机构建的虚拟机器人，所述虚拟机器人用于多次循环模拟用户操作应用程序，所述有限状态自动机包括多个状态、在所述多个状态下发生的多个事件、以及由所述多个事件触发的动作；

在每次循环中，确定所述虚拟机器人在所述应用程序中所处的视图，将所述视图作为所述状态；

在每次循环中，模拟所述用户在所述视图中选定服务，将选定的服务作为所述事件；

在每次循环中，驱动所述虚拟机器人在所述视图中执行用于触发所述服务的用户操作，将所述用户操作作为所述动作。

2、根据权利要求1所述的方法，其中，所述在每次循环中，确定所述虚拟机器人在所述应用程序中所处的视图，将所述视图作为所述状态，包括：

在首次循环中，从预置的数据库中读取在上一次循环中为所述虚拟机器人存储的、所述有限状态自动机中的状态；

在读取成功的情况下，将读取的状态配置给所述虚拟机器人，以恢复所述应用程序中的视图；

在读取失败的情况下，确定所述虚拟机器人在所述应用程序中处于默认的视图，将所述默认的视图作为所述状态。

3、根据权利要求1所述的方法，其中，所述在每次循环中，模拟所述用户在所述视图中选定服务，将选定的服务作为所述事件，包括：

在每次循环中，加载为所述视图配置的第一文件；

在所述第一文件中读取所述视图提供的服务、以及筛选规则；

模拟所述用户按照所述筛选规则选择其中一个服务，将选择的服务作为所述事件。

4、根据权利要求3所述的方法，其中，所述模拟所述用户按照所述筛选规则选择其中一个服务，将选择的服务作为所述事件，包括：

从所述筛选规则中读取为所述用户选择每个服务的概率；

按照所述概率为所述虚拟机器人选择其中一个服务，将选择的服务作为所述事件；

其中，所述服务包括闲置，所述闲置的概率大于除所述闲置外任一个服务的概率。

5、根据权利要求3所述的方法，其中，所述在每次循环中，模拟所述用户在所述视图选定服务，将选定的服务作为所述事件，还包括：

在所述视图中检测到错误，其中，所述视图提供的服务包括闲置；

响应于所述错误，模拟所述用户在所述视图选定所述闲置，将所述闲置作为所述事件。

6、根据权利要求1所述的方法，其中，所述在每次循环中，驱动所述虚拟机器人在所述视图执行用于触发所述服务的用户操作，将所述用户操作作为所述动作，包括：

在每次循环中，加载为所述视图配置的第二文件；

在所述第二文件中读取为所述服务配置的模拟规则；

驱动所述虚拟机器人在所述视图按照所述模拟规则执行所述用户操作，将所述用户操作作为所述动作。

7、根据权利要求6所述的方法，其中，所述驱动所述虚拟机器人在所述视图按照所述模拟规则执行所述用户操作，将所述用户操作作为所述动作，包括：

在所述模拟规则中读取执行所述用户操作的接口规范；

按照所述接口规范生成用于表示在所述视图执行所述用户操作的用户请求；

驱动所述虚拟机器人发送所述用户请求至服务器，作为所述动作。

8、根据权利要求6所述的方法，其中，所述驱动所述虚拟机器人在所述视图按照所述模拟规则执行所述用户操作，将所述用户操作作为所述动作，包括：

从预置的数据库中获取所述虚拟机器人在模拟所述用户操作所述应用程序的过程中记录的上下文信息；

按照所述模拟规则从所述上下文信息中读取所述服务依赖的第一环境信息；

按照所述模拟规则确定所述服务依赖的组件；

调用所述组件生成与所述第一环境信息适配的目标信息；

驱动所述虚拟机器人在所述视图发布所述目标信息，作为所述动作。

9、根据权利要求6所述的方法，其中，所述驱动所述虚拟机器人在所述视图按照所述模拟规则执行所述用户操作，将所述用户操作作为所述动作，包

括：

从预置的数据库中获取所述虚拟机器人在模拟所述用户操作所述应用程序的过程中记录的上下文信息；

按照所述模拟规则从所述上下文信息中读取限制所述服务的第二环境信息；

按照所述模拟规则确定限制所述服务的条件；

在所述第二环境信息未满足所述条件的情况下，允许驱动所述虚拟机器人在所述视图中按照所述模拟规则执行所述用户操作，作为所述动作；

在所述第二环境信息满足所述条件的情况下，禁止驱动所述虚拟机器人在所述视图中按照所述模拟规则执行用户操作，作为所述动作。

10、根据权利要求1所述的方法，其中，所述在每次循环中，驱动所述虚拟机器人在所述视图中执行用于触发所述服务的用户操作，将所述用户操作作为所述动作，包括：

在每次循环中，在所述服务为闲置的情况下，确定用于触发所述服务的用户操作为空；

响应于所述用户操作，驱动所述虚拟机器人在所述视图中等待，作为所述动作。

11、根据权利要求1-10中任一项所述的方法，还包括：

在完成执行所述用户操作的情况下，获取在所述视图中出现的上下文信息；

将所述上下文信息存储至预置的数据库中，并在所述数据库中建立所述虚拟机器人与所述上下文信息之间的关联关系。

12、根据权利要求1-10中任一项所述的方法，其中，所述有限自动状态机还包括表示所述状态变换的变换，所述方法还包括：

在完成执行所述用户操作的情况下，确定所述虚拟机器人在所述应用程序中所处的视图；

按照执行所述用户操作之后的视图更新所述状态，作为所述变换。

13、一种虚拟机器人的控制装置，包括：

虚拟机器人启动模块，设置为启动多个使用有限状态自动机构建的虚拟机器人，所述虚拟机器人用于多次循环模拟用户操作应用程序，所述有限状态自动机包括多个状态、在所述多个状态下发生的多个事件、以及由所述多个事件触发的动作；

状态确定模块，设置为在每次循环中，确定所述虚拟机器人在所述应用程序中所处的视图，将所述视图作为所述状态；

事件选定模块，设置为在每次循环中，模拟所述用户在所述视图中选定服务，将选定的服务作为所述事件；

动作执行模块，设置为在每次循环中，驱动所述虚拟机器人在所述视图中执行用于触发所述服务的用户操作，将所述用户操作作为所述动作。

14、一种虚拟机器人的控制设备，包括：

至少一个处理器；以及

与所述至少一个处理器通信连接的存储器；其中，

所述存储器存储有可被所述至少一个处理器执行的计算机程序，所述计算机程序被所述至少一个处理器执行，以使所述至少一个处理器能够执行权利要求 1-12 中任一项所述的虚拟机器人的控制方法。

15、一种计算机可读存储介质，其中，所述计算机可读存储介质存储有计算机程序，所述计算机程序用于使处理器执行时实现权利要求 1-12 中任一项所述的虚拟机器人的控制方法。

16、一种计算机程序产品，包括计算机程序，其中，所述计算机程序在被处理器执行时实现如权利要求 1-12 中任一项所述的虚拟机器人的控制方法。