与Java程序不同，Android应用程序没有主要方法。应用程序包含许多入口点，即Android框架隐式调用的方法。 Android操作系统为应用程序中的所有组件定义完整的生命周期。所有这些组件都是通过从预定义的操作系统类派生自定义类，在AndroidManifest.xml文件中注册它并覆盖生命周期方法来实现的。 Android框架调用这些方法来启动或停止组件，或者根据环境需要暂停或恢复组件。例如，它可以因内存耗尽而停止应用程序，稍后在用户返回时重新启动它。结果，在构建调用图时，Android分析不能简单地从检查预定义的“主”方法开始。相反，必须精确地对Android生命周期中的所有可能转换进行建模。为了解决这个问题，F LOW D ROID构造了一个模拟生命周期的自定义虚拟主方法。在下文中，我们将解释如何构造此方法。