

(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 104360904 A

(43) 申请公布日 2015. 02. 18

---

(21) 申请号 201410551054. 7

(22) 申请日 2014. 10. 16

(71) 申请人 四川长虹电器股份有限公司

地址 621000 四川省绵阳市高新区绵兴东路  
35 号

(72) 发明人 龚翔 程超 李奇 罗永江  
陈锡林

(74) 专利代理机构 成都虹桥专利事务所（普通  
合伙） 51124

代理人 李凌峰

(51) Int. Cl.

G06F 9/48 (2006. 01)

---

权利要求书1页 说明书2页

(54) 发明名称

基于事件总线的模块间消息传递机制

(57) 摘要

本发明涉及编程技术，特别涉及一种不同功能模块间的消息传递方法，目的是为了解决现有软件编程过程中模块间耦合严重的问题。本发明提供一种基于事件总线的模块间消息传递机制，包括如下步骤：首先，在订阅者类中实现各种事件的订阅函数并进行标注，订阅者向事件总线注册事件。其次，事件总线进行事件注册后，通过反射机制找出注册对象中被标注的订阅函数，生成订阅者对象。然后，以订阅事件为 key，用 map 来维护订阅者信息，所述订阅者信息包括订阅者对象、回调事件函数及事件类型。最后，通过反射机制分析订阅者对象，根据回调事件函数回调的 Method 名字确定事件执行方式并遍历 map 进行事件分发。本发明适用于编程领域。

1. 基于事件总线的模块间消息传递机制,其特征在于,包括如下步骤 :
  - A. 在订阅者类中实现各种事件的订阅函数并进行标注,订阅者向事件总线注册事件 ;
  - B. 事件总线进行事件注册后,通过反射机制找出注册对象中被标注的订阅函数,生成订阅者对象 ;
  - C. 以订阅事件为 key,用 map 来维护订阅者信息,所述订阅者信息包括订阅者对象、回调事件函数及事件类型 ;
  - D. 通过反射机制分析订阅者对象,根据回调事件函数回调的 Method 名字确定事件执行方式并遍历 map 进行事件分发。
2. 如权利要求 1 所述的基于事件总线的模块间消息传递机制,其特征在于,步骤 D 中事件执行方式包括 :

PostThread: 直接调用订阅对象的回调函数 ;

MainThread: UI 线程的 Handler, PendingPostQueue 来维护一个事件列表, SendMessage 方式发送消息到 UI Looper, 设定超时机制, 如果事件处理超时, 会将后续的事件重新发送一次, 防止阻塞 UI 线程 ;

Async : 实现 Runnable 接口, 维护一个 PendingPostQueue 事件队列, 执行在事件总线创建的线程池中 ;

BackgroundThread: 判断当前线程是否是主线程, 如果是, 执行方式与 PostThread 一致, 如果当前线程不是主线程, 则维护一个 PendingPostQueue 事件队列, 并执行在事件总线创建的线程池中。
3. 如权利要求 1 所述的基于事件总线的模块间消息传递机制,其特征在于,步骤 A 中以 @Subscribe 标注订阅函数。
4. 如权利要求 1 所述的基于事件总线的模块间消息传递机制,其特征在于,所述反射机制为 java 反射机制。

## 基于事件总线的模块间消息传递机制

### 技术领域

[0001] 本发明涉及编程技术,特别涉及一种不同功能模块间的消息传递方法。

### 背景技术

[0002] 在应用开发过程中,我们总会遇到各个组件模块之间的通信,随着应用功能的增加,需要监听的事件越来越多,越来越多的模块需要监听不同的事件,导致大量的模块需要注册到各事件管理器上,从而出现各模块间耦合严重。代码不易维护和测试,甚至在出现一些 bug 的时候,会无从入手。

[0003] 软件产品的终极目标是松耦合,模块之间应该尽量减少耦合度,不要违背软件工程中“高内聚,低耦合”的原则。很多技术的出现都是为解决耦合问题,比如 IoC、AOP、SOA、观察者模式、面向接口编程、MVC、事件驱动架构(Event-Driven Architecture)等方法。

### 发明内容

[0004] 本发明的目的是为了解决现有软件编程过程中模块间耦合严重的问题。

[0005] 本发明提供一种基于事件总线的模块间消息传递机制,包括如下步骤:

[0006] A. 在订阅者类中实现各种事件的订阅函数并进行标注,订阅者向事件总线注册事件;

[0007] B. 事件总线进行事件注册后,通过反射机制找出注册对象中被标注的订阅函数,生成订阅者对象;

[0008] C. 以订阅事件为 key,用 map 来维护订阅者信息,所述订阅者信息包括订阅者对象、回调事件函数及事件类型;

[0009] D. 通过反射机制分析订阅者对象,根据回调事件函数回调的 Method 名字确定事件执行方式并遍历 map 进行事件分发。

[0010] 具体地,步骤 D 中事件执行方式包括:

[0011] PostThread: 直接调用订阅对象的回调函数;

[0012] MainThread: UI 线程的 Handler, PendingPostQueue 来维护一个事件列表,SendMessage 方式发送消息到 UI Looper, 设定超时机制, 如果事件处理超时, 会将后续的事件重新发送一次, 防止阻塞 UI 线程;

[0013] Async: 实现 Runnable 接口, 维护一个 PendingPostQueue 事件队列, 执行在事件总线创建的线程池中;

[0014] BackgroundThread: 判断当前线程是否是主线程, 如果是, 执行方式与 PostThread 一致, 如果当前线程不是主线程, 则维护一个 PendingPostQueue 事件队列, 并执行在事件总线创建的线程池中。

[0015] 优选地, 步骤 A 中以 @Subscribe 标注订阅函数。

[0016] 具体地, 所述反射机制为 java 反射机制。

[0017] 本发明的有益效果是: 本发明基于对 Publish/Subscribe 模式的改进优化, 提出

事件总线概念,减少创建复杂的类和接口层次结构使用以增强代码的健壮性和可靠性。

## 具体实施方式

[0018] 本发明针对现有软件编程过程中模块间耦合严重的问题,提出一种基于事件总线的模块间消息传递机制,包括如下步骤:

[0019] A. 在订阅者类中实现各种事件的订阅函数并进行标注,订阅者向事件总线注册事件;

[0020] B. 事件总线进行事件注册后,通过反射机制找出注册对象中被标注的订阅函数,生成订阅者对象;

[0021] C. 以订阅事件为 key,用 map 来维护订阅者信息,所述订阅者信息包括订阅者对象、回调事件函数及事件类型;

[0022] D. 通过反射机制分析订阅者对象,根据回调事件函数回调的 Method 名字确定事件执行方式并遍历 map 进行事件分发。

[0023] 事件总线存在如下技术特点:

[0024] 1、利用 java 反射机制实现订阅事件函数基于注解 (annotation)。

[0025] 2、事件响应有多线程选择,具体包括如下几种方式:

[0026] PostThread :事件响应函数和事件发布在同一线程中执行,这样可以避免线程切换。

[0027] MainThread :事件响应函数会在主线程 (UI 线程) 中执行。

[0028] BackgroundThread :事件响应函数会在一个后台线程中执行。如果事件发布函数不是在主线程中,则会立即在事件发布线程中执行响应函数。如果事件发布函数在主线程中,事件总线则会在唯一的一个后台线程中按照顺序来执行所有的后台事件响应函数。

[0029] Async :事件响应函数在另外一个异步线程中执行。该线程和发布线程、主线程相互独立。处理事件响应函数需要较长的时间来执行。

[0030] BackgroundThread: 判断当前线程是否是主线程,如果是,执行方式与 PostThread 一致,如果当前线程不是主线程,则维护一个 PendPostQueue 事件队列,并执行在事件总线创建的线程池中。