

# COMP3027J

## Software Architecture

### Software Architectural Styles (Event Systems Style)

DENG, YONGJIAN

Faculty of Computer Science, BJUT

Data Mining & Security Lab (DMS Lab)



# Outline

1. What is Event
2. Event Systems Architectural Style
3. Dispatch Mechanism of Event Systems
4. Advantages and Disadvantages



北京工业大学  
BEIJING UNIVERSITY OF TECHNOLOGY

# Outline

1. What is Event
2. Event Systems Architectural Style
3. Dispatch Mechanism of Event Systems
4. Advantages and Disadvantages



# Event



- **Events are actions** that activate the functionality of an object. When such an action occurs, a **message** is sent to the object involved, and the object can perform the **corresponding function**
- When an event occurs to an object, the object finds a corresponding program that handles the event to handle the event



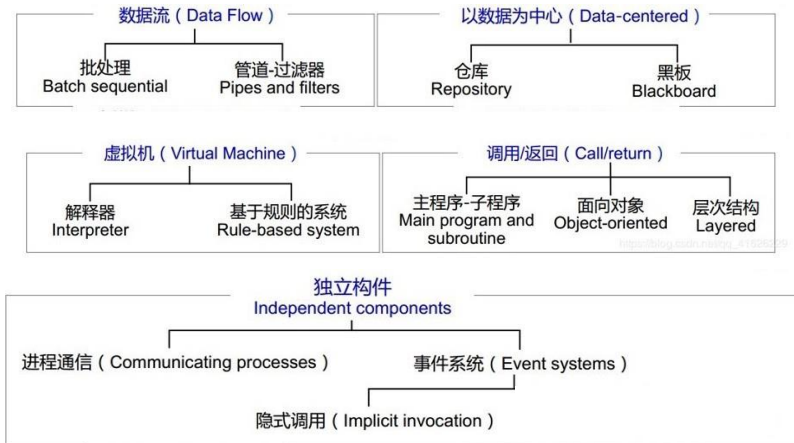
# Outline

1. What is Event
2. Event Systems Architectural Style
3. Dispatch Mechanism of Event Systems
4. Advantages and Disadvantages



北京工业大学  
BEIJING UNIVERSITY OF TECHNOLOGY

# Taxonomy of styles



# Characteristics of Event Systems-Implicit Invocation

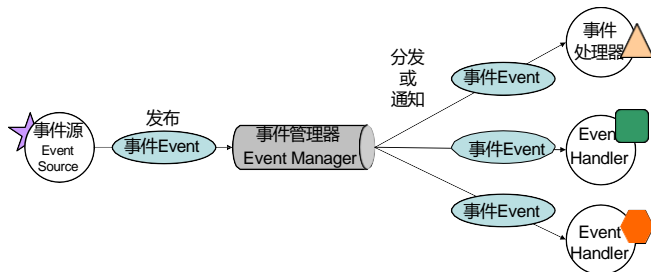


- The triggerer of the event does not know which components will be affected by these events and **remains independent of each other**.
- **The order in which components are processed cannot be assumed**, or even which procedures will be called.
- Each component has no connection relationship with each other and exists independently, and **the association is realized by publishing and registering events**.



# Characteristics of Event Systems-Implicit Invocation

这个图要记



Characteristics	Description
Separate interaction	Event publishers are not aware of the existence of event subscribers.
One to many communication	With publish/subscribe messaging, a particular event can affect multiple subscribers.
Event-based trigger	Called by event-triggered procedures.
Asynchronous	Asynchronous operations are supported





**In an implicit-invocation-style system, an event implicitly causes the invocation of procedures in other modules.**

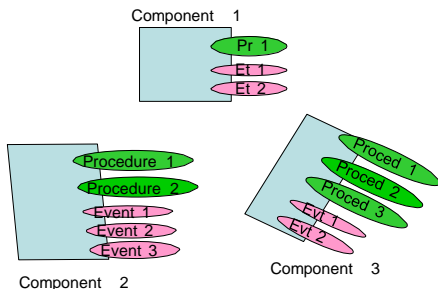
- A component(**Event Source**) can announce (or **Broadcast**) one or more events. 一个组件可以广播一些事件。
- Other components(**Event Handlers**) in the system can register an interest in an event by associating a procedure with it. 系统中的其它构件可以注册自己感兴趣的事件，并将自己的某个过程与相应的事件进行关联。
- When the event is announced, the system (**Event Manager**) itself invokes all of the procedures that have been registered for the event. 当一个事件被发布，系统自动调用在该事件中注册的所有过程。



# Components of Event Systems Style

**Components of event systems: objects or processes whose interfaces provide both:**

- a collection of procedures (过程或函数, 充当事件源或事件处理器的角色)
- a set of events (事件)



## Connectors of event systems: event-procedure bindings

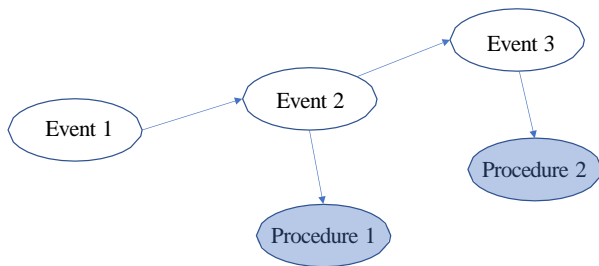
- Procedures are registered with events (过程 < 事件处理器 > 向特定的事件进行注册)
- Components communicate by announcing events at “appropriate” times (构件 < 事件源 > 发布事件)
- when an event is announced the associated procedures are (implicitly) invoked (当某些事件被发布时，向其注册的过程被隐式调用)
- Order of invocation is non-deterministic (调用的次序是不确定的)



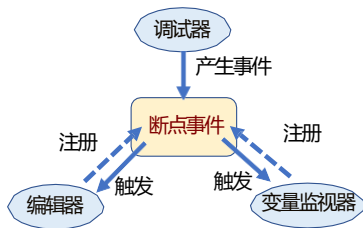
# Connectors of Event Systems Style

## Connectors of event systems: event-procedure bindings

- In some treatments, connectors are event-event bindings (在某些情况下，一个事件也可能触发其他事件，形成事件链).



# Examples of Event Systems Style



► **Event Source:** debugger (调试器)

► **Event Handler:** editor and variable monitor (编辑器与变量监视器)

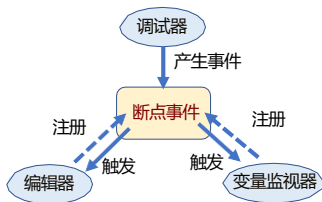
► **Event Manager:** IDE (集成开发环境)

## How?



北京工业大学  
BEIJING UNIVERSITY OF TECHNOLOGY

# Examples of Event Systems Style



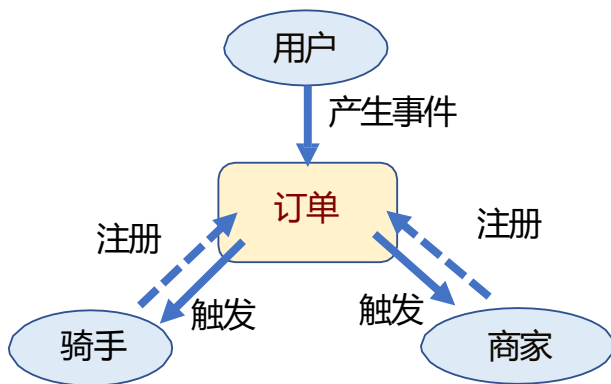
► **Event Source:** debugger (调试器)

► **Event Handler:** editor and variable monitor (编辑器与变量监视器)

► **Event Manager:** IDE (集成开发环境)

- The editor and variable monitor register with the debugger to receive "breakpoint events";
- When a breakpoint is encountered, the debugger publishes an event, which triggers the "editor" and "variable watcher"
- The editor scrolls the source code to the breakpoint;
- The variable watcher updates the current variable value and displays it.

# Examples of Event Systems Style



# Outline

1. What is Event
2. Event Systems Architectural Style
3. Dispatch Mechanism of Event Systems
4. Advantages and Disadvantages





# Dispatch mechanism of Event Systems

**When an event occurs, the procedures registered with this event are fired and executed. In what way will the event be dispatched to the registered process? How will the process be performed?**

- EventManager w/o dispatcher module (无独立调度派遣模块的事件管理器)
- EventManager w/ dispatcher module (带有独立派遣模块的事件管理器)



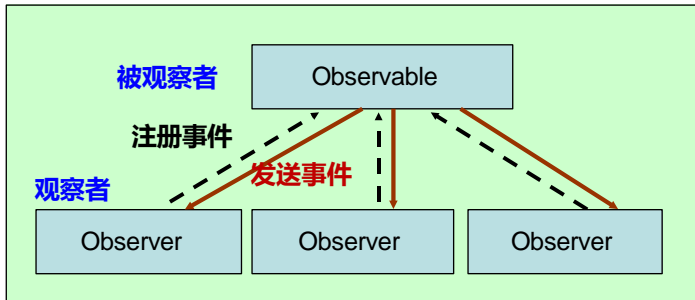
# Event Systems w/o Dispatcher Module

- This module is usually called Observable/Observer (被观察者/观察者).
- Each module allows other modules to declare interest in events that they are sending. (每一个模块都允许其他模块向自己所能发送的某些消息表明兴趣)
- Whenever a module sends an event, it sends that event to exactly those modules that registered interest in that event. (当某一模块发出某一事件时, 它自动将这些事件发布给那些曾经向自己注册过此事件的模块)



# Event Systems w/o Dispatcher Module

## Observable/Observer Style



Observable/Observer module

.....► Register Event

————► Send event



# Event Systems w/ Dispatcher Module

## What?

**The dispatcher module** is responsible for receiving all incoming events and dispatching them to other modules in the system.

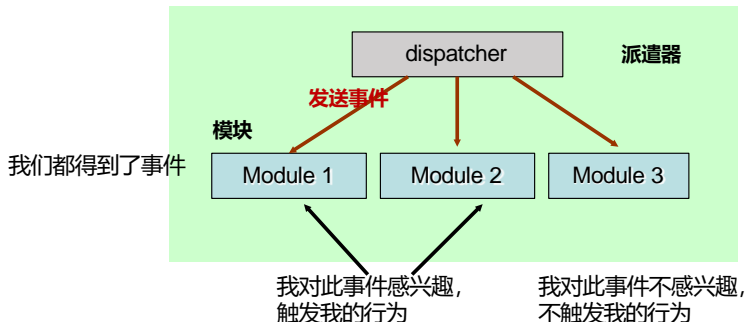
## How?

- All broadcasting: The dispatch module broadcasts the event to all modules, but only interested modules fetch the event and trigger their own behavior.
- Selected broadcasting: The dispatch module sends events to those registered modules.



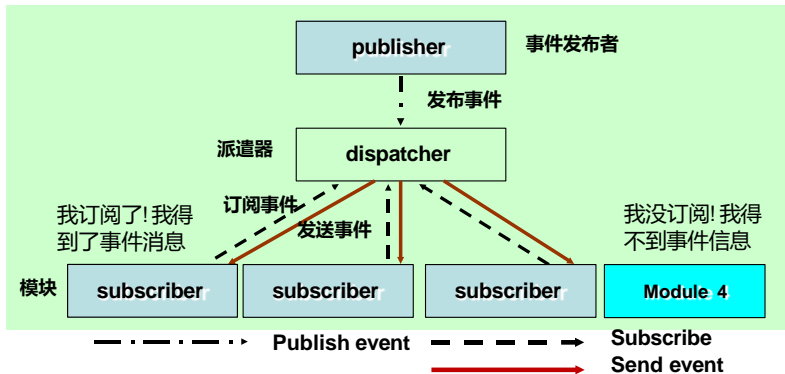
# Event Systems w/ Dispatcher Module

## All broadcasting



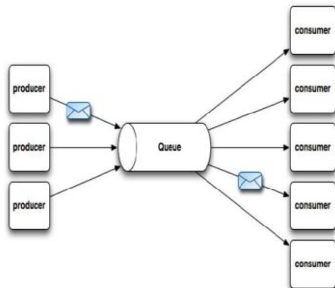
# Event Systems w/ Dispatcher Module

## Selected broadcasting

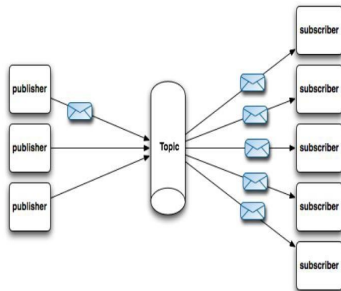


# Two Strategies of Selected Broadcasting

## Point-to-Point (message queue)

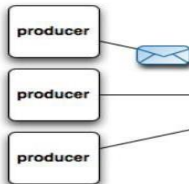


## Publish-Subscribe



# Point-to-Point (message queue)

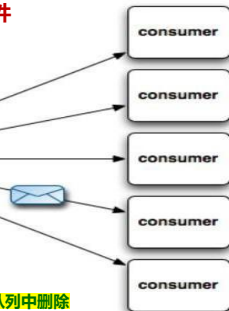
其他的应用连接到  
该队列并向其中发  
布事件



系统配置一个队列管理器，  
并定义一个命名的消息队列

队列管理器存储这些消息，直到  
接收端的应用连接到队列，取回  
这些消息并加以处理

某个应用向消息队列注册，以  
监听并处理被放置在队列里的  
事件



消息只能被唯一的消费者所消费，消费之后即从队列中删除



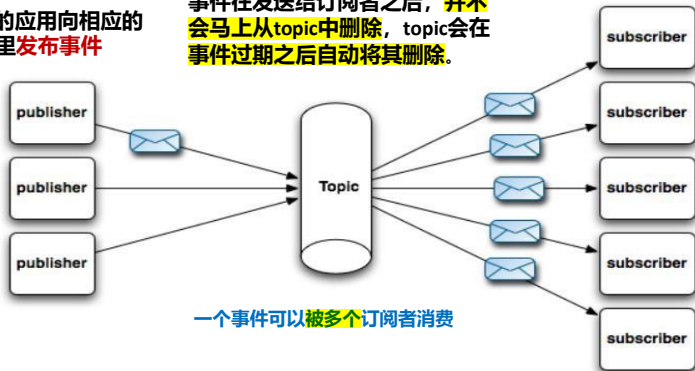


# Publish-Subscribe

某个应用向topic注册，以监听并处理被放置在topic里的事件

其他的应用向相应的topic里发布事件

事件在发送给订阅者之后，并不会马上从topic中删除，topic会在事件过期之后自动将其删除。



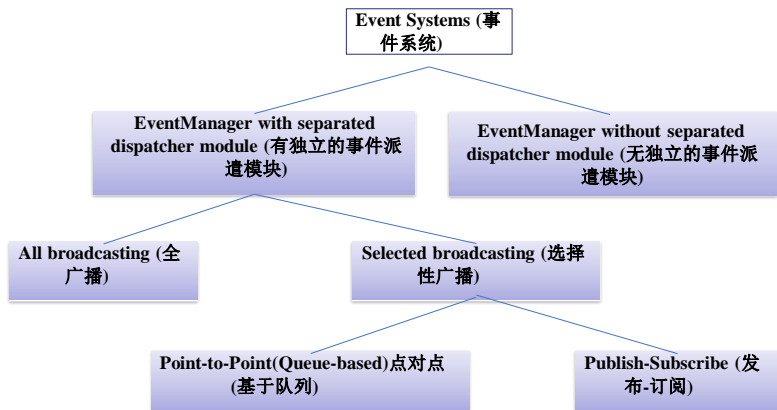
一个事件可以被多个订阅者消费



北京工业大学  
BEIJING UNIVERSITY OF TECHNOLOGY

# Categorization of Event Systems

有独立，无独立的区别



北京工业大学  
BEIJING UNIVERSITY OF TECHNOLOGY

# Outline

1. What is Event
2. Event Systems Architectural Style
3. Dispatch Mechanism of Event Systems
4. Advantages and Disadvantages



# Advantages of Event Systems Style



1. The event manager and the event handlers are separate, providing decoupled and individual functionalities.
2. The replacement and additions are independent and thus easier to perform.
3. Any manager or handler malfunction will not affect the other managers and handlers.
4. High reuse potential.



# Disadvantages of Event Systems Style



1. It is difficult for the dispatcher to react to a myriad of inputs and respond in time (especially on simultaneous inputs).
2. A dispatcher malfunction will bring the whole system down.
3. Dispatcher is the performance bottleneck. It must be fast.
4. No insurance that an event will be treated.



# Thank you!



北京工业大学  
BEIJING UNIVERSITY OF TECHNOLOGY