Events

VISUALIZATION ON THE WEB

HTML DOM Events

- DOM Events allow to handle interaction with the page
- An event is registered with a listener, usually a function
- When an event e occurs, the corresponding listener is executed

Event type

- Dom can generate different types of events

- A selection of the most frequently used
  - onclick
  - onmouseover
  - onmouseout
  - onsubmit

D3.js events and listeners

- D3.js provides a set of utility functions to manage events
- `d3.selection.on(type[,listener[,capture]])`
  - `type` is a string to refer to a specific event type
    - E.g.: “mousedown”,”mouseover”,”click”
  - The `listener` function is invoked as an operator function, passing current datum and index
  - If listener is not specified, returns the list of listeners for the given type
  - If listener is null, the listener is removed for that type
  - To register multiple listeners for the same type, the type specifier may be followed by an optional namespace:
    - Eg: “click.bar”, “click.foo”

Example Geo Paintings

d3.dispatch

- Dispatching allows a flexible management of events, in scenarios with complex dependencies among components
- It allows to define custom events, with customizable parameters
  - `var dispatch = d3.dispatch(“start”,”end”)`
- Listerners can be attached to the dispatcher
  - `dispatch.on(“start”, function(d,i){…})`
  - `dispatch.on(“start.foo”, function(d,i){…})`
  - `dispatch.on(“end”, function(d,i){…})`
- Emit an event
  - `dispatch.start(argument1, argument2)`
  - `dispatch.end()`

Example Vast 2008