

GISpL: the Gestural Interface Specification Language

draft-echtler-gispl-specification-01

Florian Echtler <floe@butterbrot.org>

March 28th, 2011

IETF Meeting 80, Prague

Motivation

The rise of gesture-based input

- research topic since early 1980s („VIDEOPLACE“)
- entering mainstream since ~ 2005
- examples:
 - touchscreens/multitouch surfaces (tablets, interactive tables)
 - tangible interfaces (Reactable)
 - free-air gestures (Wiimote, Kinect)
- current issues:
 - many slightly different ad-hoc implementations
 - hard-coded behavior
 - no customization

Motivation

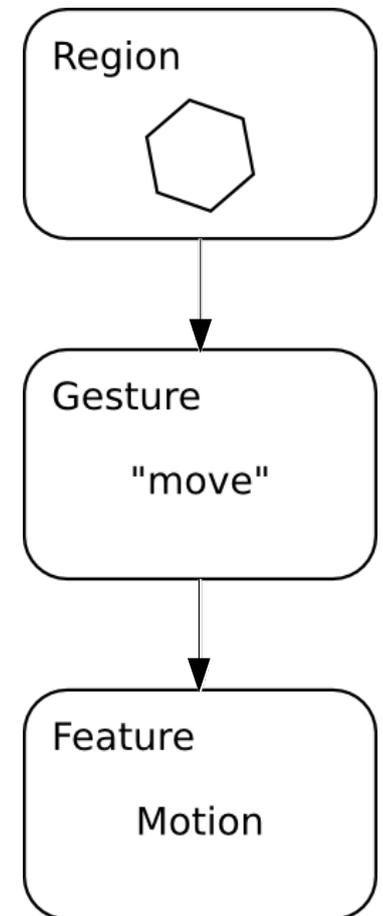
What is a gesture, and how can I describe it to a computer?

- solution approach:
hardware-independent, machine-readable gesture descriptions
- unified gesture descriptions allow...
 - faster development
 - user customization
 - improved portability
- hard question: *what is a gesture?*
- answer within this context:
any motion(s) which the user executes to achieve a certain effect in the UI
- directly leads to the next question:
how can these motions be described?

Core Concepts

Abstract Description of Gestures

- Regions:
 - spatial areas defined in reference coordinate system
 - extension of „traditional“ UI windows
- Gestures:
 - sequences of features, either ...
 - pre-defined by capability description or ...
 - customized by application
- Features:
 - geometrical/mathematical properties of input data, e.g.:
 - motion vector
 - relative rotation
 - travelled path
 - further classification through filters and constraint values



Example

Composite Gesture

GISpL: based on JSON → human- & machine-readable

example: horizontal swipe with two fingers

```
{
  "name": "two_finger_swipe",
  "flags": "oneshot",
  "features": [
    {
      "type": "Count",
      "filters": 2,
      "constraints": [2, 2],
      "result": []
    }, {
      "type": "Motion",
      "filters": 2,
      "constraints": [[100, 0, 0], [10000, 10, 10]],
      "result": []
    }
  ]
}
```

Result (only when constraints match): (3-vector, integer) = motion vector + object count

Thank you for your attention!

Questions & comments?

Core Concepts

Examples of available features

- Motion – average motion vector
- Rotation – rotation around center of mass
- Scale – scaling w.r.t. center of mass
- Path – recognize „shape-based“ gestures

- ObjectCount – number of objects inside region
- ObjectDimensions – describes shape of object
- ObjectOrientation – rotation relative to reference frame
- ObjectPosition – absolute position of object
- ObjectID – unique ID (e.g., fiducial marker) of an object