The Big Board.

Jeff Heard
The Renaissance Computing Institute

jeff@renci.org
http://vis.renci.org/jeff

The problem.

- Disasters don't obey state or county lines nor resource constraints.
- Thus resources for one disaster need to be coordinated effectively across boundaries.
- EMs need a geographically coordinated shared workspace.
- Most natural shared workspace is a map.

Existing Solutions.

- Google Maps / Earth
  - Content is static
- ArcInfo
  - Solutions are difficult to integrate
- WebEOC
  - Non georeferenced
- Pen and Paper
  - Need I really say more?

The Big Board.

- Teleconferencing over maps
- Realtime content creation and generation
- Orthophotography or vector layers
- 100% Haskell with Python server-side.
- Directly uses Hieroglyph, buster, haxr (xml-rpc), mtl (applicative), gtk2hs, parsec, HTTP 4000.x, bytestring.

Why Haskell?

- Memory and speed efficient executables.
- Excellent graphics libraries.
- Reliability through type-safety.
- Rapid development.
- High level of code reuse.

FP vs. Traditional vis.

- Visual Semantics
  - Visual meaning vs. data and purpose of application.
- Abstracts away from process of drawing.
- Ties visual representation to data closely.

Interactivity via FRP.

- Functional Reactive Programming models interactivity via behaviours and events.
- Behaviours react to events.
- Buster, TBB's FRP system is a broadcast FRP as opposed to an arrow-based FRP.

Postmortem.

- The Big Board:
  - Less than 1000 lines of app-specific code.
- Two libraries, plus the beginnings of a third.
  - Hieroglyph for pure-functional vis graphics.
  - Buster for “app-orchestration”
  - Beginnings of GIS library for parsing WKT, WKB, PROJ, and interfacing with libproj2

Advantages of FP.

- Rapid development.
- Encouragement of reuse and small understandable codebase.
- High performance compared to other high-level languages.
- Active and responsive dev community on #haskell and [Haskell-cafe].

Disadvantages of FP.

- Relatively small dev community,
- Dev community overlap with the graphics community is not yet all that large.
- Much of FP is still theory-focused.
- Few colleagues at my office understand FP or think they have time to learn.

Lessons Learned.

- Shortcuts are generally the long way around.
- The type system should be used to concentrate on semantics rather than modularization.
- “pure” functions preferable to IO functions, because it encourages deeper thought about the problem. More reuse, clearer code.