

Refactoring in the 4th Dimension

Michael Feathers
Chief Scientist, Obtiva Corporation

One Data Structure..

CodeEvent:

- commit # sha1 from git
- date # commit date
- committer # person who did the deed
- file_name
- class_name
- method_name
- complexity # flog for method

**It's important to remember
that every event is for a
method**

Load a Repository

```
events = read_events( 'diaspora' )
```

Get all of the file names

```
events.map { |e| e.file_name }
```

..if you want to see them :-)

```
ap events.map { |e| e.file_name }
```

Sorted list of file names

```
events.map { |e| e.file_name }.sort.uniq
```

Let's get some class names

```
events.map { |e| e.class_name }.sort.uniq
```

What about committers?

```
events.map { |e| e.committer }.sort.uniq
```

Let's see how many
commits Mike Hill has..

```
events.select { |e| e.committer == "Hill" }  
        .count
```

Let's see what time of day he commits

```
events.select { |e| e.committer == "Hill" }  
      .map { |e| e.date.hour }  
      .freq
```

Last commit in the code

```
events.date_sorted.last
```

Highest method complexity over time

```
events.map { |e| e.complexity }.max
```

..to get the name too

```
max_event = events  
  .sort {|left,right| left.complexity <=> right.complexity }  
  .last
```

```
ap max_event.method_name  
ap max_event.complexity
```

Number of commits per file

```
events.group_by { |e| e.file_name }  
      .map { |name, events| [name, events.count] }
```

Break up your work..

```
method_names = events.map { |e| e.method_name }.sort.uniq
```

```
ap method_names
```

```
events.date_sorted.select { |e| e.method_name == "initialize" }  
                    .last  
                    .complexity
```

Go to it!