Racket is my Mjolnir

(sixth RacketCon)

18 September 2016

Geoffrey Knauth

Who am I?

Geoff Knauth, programming since 1975 (age 15).

Who am I?

Geoff Knauth, programming since 1975 (age 15).

- When I was in high school, we thought we knew everything.

Who am I?

Geoff Knauth, programming since 1975 (age 15).

- When I was in high school, we thought we knew everything.
- *Well*... we didn't know who Don Knuth was.
- We didn't know who John McCarthy was.
- We'd never heard of Alan Turing or Alonzo Church.

Who am I?

Geoff Knauth, programming since 1975 (age 15).

- When I was in high school, we thought we knew everything.
- *Well*... we didn't know who Don Knuth was.
- We didn't know who John McCarthy was.
- We'd never heard of Alan Turing or Alonzo Church.

Basically, we knew a lot less than we thought we did.

Then you grow up, you learn, you meet people...

Teachers, mentors & friends introduce you to more:

Harry Lewis, Hartley Rogers, Richard Stallman, Marvin Minsky, Gerry Sussman, Hal Abelson, Radia Perlman, Dan Bricklin, Margo Seltzer, Steve Jobs, Don Knuth, Alan Kay, Dennis Ritchie, Ken Thompson, Linus Torvalds, Guy Steele, Ray Tomlinson, Rusty Bobrow, Ken Anderson, Peter Norvig, Gregor Kiczales, Brian Kernighan, Shriram Krishnamurthi, Jeremy Hylton, Matthias Felleisen, Philip Wadler, Vint Cerf, Matthew Flatt, Andy van Dam, John McCarthy, Robby Findler, John Nash, Michael Rabin, Leslie Valiant, Adi Shamir, Dan Friedman, Stephen Wolfram, Gil Strang, ...

choronological order from the 1970s

Then you grow up, you learn, you meet people...

Teachers, mentors & friends introduce you to more:

Harry Lewis, Hartley Rogers, **Richard Stallman**, **Marvin Minsky**, **Gerry Sussman**, **Hal Abelson**, Radia Perlman, Dan Bricklin, Margo Seltzer, Steve Jobs, Don Knuth, Alan Kay, Dennis Ritchie, Ken Thompson, Linus Torvalds, **Guy Steele**, **Ray Tomlinson**, **Rusty Bobrow**, **Ken Anderson**, **Peter Norvig**, **Gregor Kiczales**, Brian Kernighan, **Shriram Krishnamurthi**, Jeremy Hylton, **Matthias Felleisen**, **Philip Wadler**, Vint Cerf, **Matthew Flatt**, Andy van Dam, **John McCarthy**, **Robby Findler**, John Nash, Michael Rabin, Leslie Valiant, Adi Shamir, **Dan Friedman**, Stephen Wolfram, Gil Strang, ...

Yea Lisp!

Then you grow up, you learn, you meet people...

Teachers, mentors & friends introduce you to more:

Harry Lewis, Hartley Rogers, Richard Stallman, Marvin Minsky, Gerry Sussman, Hal Abelson, Radia Perlman, Dan Bricklin, Margo Seltzer, Steve Jobs, Don Knuth, Alan Kay, Dennis Ritchie, Ken Thompson, Linus Torvalds, Guy Steele, Ray Tomlinson, Rusty Bobrow, Ken Anderson, Peter Norvig, Gregor Kiczales, Brian Kernighan, Shriram Krishnamurthi, Jeremy Hylton, Matthias Felleisen, Philip Wadler, Vint Cerf, Matthew Flatt, Andy van Dam, John McCarthy, Robby Findler, John Nash, Michael Rabin, Leslie Valiant, Adi Shamir, Dan Friedman, Stephen Wolfram, Gil Strang, ...

So be nice to your teachers, be nice to everyone.

Then you grow up, you learn, you meet people...

Teachers, mentors & friends introduce you to more:

Harry Lewis, Hartley Rogers, Richard Stallman, Marvin Minsky, Gerry Sussman, Hal Abelson, Radia Perlman, Dan Bricklin, Margo Seltzer, Steve Jobs, Don Knuth, Alan Kay, Dennis Ritchie, Ken Thompson, Linus Torvalds, Guy Steele, Ray Tomlinson, Rusty Bobrow, Ken Anderson, Peter Norvig, Gregor Kiczales, Brian Kernighan, Shriram Krishnamurthi, Jeremy Hylton, Matthias Felleisen, Philip Wadler, Vint Cerf, Matthew Flatt, Andy van Dam, John McCarthy, Robby Findler, John Nash, Michael Rabin, Leslie Valiant, Adi Shamir, Dan Friedman, Stephen Wolfram, Gil Strang, ...

So be nice to your teachers, be nice to everyone.
There's so much out there, and so many great people.

Especially you, PLT!

Caveat

I am a Senior Software Developer at AccuWeather (since March 2014).

I do not speak for AccuWeather.

This talk and project are separate from AccuWeather. I saw a **public weather data format** that needed better tooling and made this a side project.

Let me tell you about AccuWeather: they know as much about meteorology as you know about program language design.

I am not a meteorologist. But I've been a pilot since 1978, and I often fly through weather, though I try really hard to avoid the bad stuff, because it will kill you (fog, ice, thunderstorms).

I get to work with lots of weather data

Hundreds of NWS and NOAA message formats, dozens of (ever-changing) formats from other countries in dozens of different languages, all the info you'd expect plus volcanic eruptions and ash clouds, tornadoes, tropical cyclones, aviation, flooding, air quality, individual enthusiast reporting stations, radar imagery, atmospheric measurements at all altitudes, chemistry, ...

What is Mjolnir?

What is Mjolnir?

What is Mjolnir?

"Mjolnir is forged by Dwarven blacksmiths, and is composed of the fictional Asgardian metal uru. The side of the hammer carries the inscription..."

What is Mjolnir?

"Mjolnir is forged by Dwarven blacksmiths, and is composed of the fictional Asgardian metal uru. The side of the hammer carries the inscription..."

Whosoever holds this hammer, if he be worthy, shall possess the power of Thor.

Whosoever holds this hammer, if he be worthy, shall possess the power of Thor.

Whosoever holds this hammer, if he be worthy, shall possess the power of Thor.

Note: Thor can also be female.

Whosoever holds this hammer, if he be worthy, shall possess the power of Thor.

Note: Thor can also be female.

Be worthy.

Be worthy.

Use Racket.

So what's this talk really about?

Racket and Weather

So what's this talk really about?

Ok, that was a bit broad.

So what's this talk really about?

Racket and WMO BUFR

So what's this talk really about?

Racket and WMO BUFR

Using Racket with BUFR files

What's a BUFR file?

Binary Universal Form for the Representation of meteorological data

BUFR Files Big Concept

- BUFR consolates disparate weather data formats into one.
- It dates from the 1980s, when modems were slow, and every bit counted.
  - Every bit still counts where communications infrastructure is old or slow.

BUFR Files Infrastructure

- The original software was built in Fortran.
- C interfaces evolved. More recently, Java & Python.
- Most programs that deal with BUFR files are **compiled**.
  - And that's what drove this effort to use Racket.

Gift is something else in another language

I was given some BUFR files.

I decoded them with several standard decoders.

The output was **not pretty.**

I looked at the output. *Is this COBOL?*

I tried JSON output. *What the ...?*

Example BUFR decoder text output

```
DATA DESCRIPTORS  1 001033  2 001034
 3 001032  4 001025  5 001027  6 001090
 7 001091  8 001092  9 301011 10 301012
11 008005 12 005002 13 006002 14 008005
15 005002 16 006002 17 010051 18 008005
19 301023 20 011012 21 108000 22 031001
23 008021 24 004024 25 008005 26 301023
27 010051 28 008005 29 301023 30 011012
```

BUFR describes many kinds of weather data

- 7650 descriptors
  - 5152 unique descriptors
- more than 66 ways to describe time
- more than 51 ways to describe wind

Example BUFR decoder text output

```
DATA DESCRIPTORS  1 001033  2 001034
 3 001032  4 001025  5 001027  6 001090
 7 001091  8 001092  9 301011 10 301012
11 008005 12 005002 13 006002 14 008005
15 005002 16 006002 17 010051 18 008005
19 301023 20 011012 21 108000 22 031001
23 008021 24 004024 25 008005 26 301023
27 010051 28 008005 29 301023 30 011012
```

Example BUFR decoder text output

```
  1  001033  IDENTIFICATION OF ORIGINATING/GENERATING CENTRE
  2  001034  IDENTIFICATION OF ORIGINATING/GENERATING SUB-CENTRE
  3  001032  GENERATING APPLICATION
  4  001025  STORM IDENTIFIER
  5  001027  WMO LONG STORM NAME
  6  001090  TECHNIQUE FOR MAKING UP INITIAL PERTURBATIONS
  7  001091  ENSEMBLE MEMBER NUMBER
  8  001092  TYPE OF ENSEMBLE FORECAST
  9  004001  YEAR  10  004002  MONTH
 11  004003  DAY  12  004004  HOUR
 13  004005  MINUTE
 14  008005  METEOROLOGICAL ATTRIBUTE SIGNIFICANCE
 15  005002  LATITUDE (COARSE ACCURACY)
 16  006002  LONGITUDE (COARSE ACCURACY)
 17  008005  METEOROLOGICAL ATTRIBUTE SIGNIFICANCE
 18  005002  LATITUDE (COARSE ACCURACY)
 19  006002  LONGITUDE (COARSE ACCURACY)
 20  010051  PRESSURE REDUCED TO MEAN SEA LEVEL
 21  008005  METEOROLOGICAL ATTRIBUTE SIGNIFICANCE
 22  005002  LATITUDE (COARSE ACCURACY)
 23  006002  LONGITUDE (COARSE ACCURACY)
 24  011012  WIND SPEED AT 10 M
 25  031001  DELAYED DESCRIPTOR REPLICATION FACTOR
...  ......  .....
416  008021  TIME SIGNIFICANCE
417  004024  TIME PERIOD OR DISPLACEMENT
418  008005  METEOROLOGICAL ATTRIBUTE SIGNIFICANCE
419  005002  LATITUDE (COARSE ACCURACY)
420  006002  LONGITUDE (COARSE ACCURACY)
421  010051  PRESSURE REDUCED TO MEAN SEA LEVEL
422  008005  METEOROLOGICAL ATTRIBUTE SIGNIFICANCE
423  005002  LATITUDE (COARSE ACCURACY)
424  006002  LONGITUDE (COARSE ACCURACY)
425  011012  WIND SPEED AT 10 M
```

Example BUFR decoder text output

```
  1 IDENTIFICATION OF ORIGINATING/GE   0.98000000000000E+002 COMMONCODETABLEC-1
  2 IDENTIFICATION OF ORIGINATING/GE                 MISSING COMMONCODETABLEC-12
  3 GENERATING APPLICATION             0.10000000000000E+001 CODETABLEDEFINEDBYORIGIN
  4 STORM IDENTIFIER                   0.10030000000000E+004 CCITTIA5                04L
  5 WMO LONG STORM NAME                0.53010000000000E+005 CCITTIA5                     DANNY
  6 TECHNIQUE FOR MAKING UP INITIAL    0.20000000000000E+001 CODE TABLE 1090
  7 ENSEMBLE MEMBER NUMBER             0.10000000000000E+001 NUMERIC
  8 TYPE OF ENSEMBLE FORECAST                        MISSING CODE TABLE 1092
  9 YEAR                               0.20150000000000E+004 A
 10 MONTH                              0.80000000000000E+001 MON
 11 DAY                                0.19000000000000E+002 D
 12 HOUR                               0.00000000000000E+000 H
 13 MINUTE                             0.00000000000000E+000 MIN
 14 METEOROLOGICAL ATTRIBUTE SIGNIFI   0.10000000000000E+001 CODE TABLE 8005
 15 LATITUDE (COARSE ACCURACY)         0.11100000000000E+002 DEG
 16 LONGITUDE (COARSE ACCURACY)       -0.38300000000000E+002 DEG
 17 METEOROLOGICAL ATTRIBUTE SIGNIFI   0.40000000000000E+001 CODE TABLE 8005
 18 LATITUDE (COARSE ACCURACY)         0.11300000000000E+002 DEG
 19 LONGITUDE (COARSE ACCURACY)       -0.38600000000000E+002 DEG
 20 PRESSURE REDUCED TO MEAN SEA LEV   0.10090000000000E+006 PA
 21 METEOROLOGICAL ATTRIBUTE SIGNIFI   0.30000000000000E+001 CODE TABLE 8005
 22 LATITUDE (COARSE ACCURACY)         0.99000000000000E+001 DEG
 23 LONGITUDE (COARSE ACCURACY)       -0.38000000000000E+002 DEG
 24 WIND SPEED AT 10 M                 0.19000000000000E+002 M/S
 25 DELAYED DESCRIPTOR REPLICATION F   0.40000000000000E+002 NUMERIC
... .....                              .....
416 TIME SIGNIFICANCE                              MISSING CODE TABLE 8021
417 TIME PERIOD OR DISPLACEMENT                    MISSING H
418 METEOROLOGICAL ATTRIBUTE SIGNIFI               MISSING CODE TABLE 8005
419 LATITUDE (COARSE ACCURACY)                     MISSING DEG
420 LONGITUDE (COARSE ACCURACY)                    MISSING DEG
421 PRESSURE REDUCED TO MEAN SEA LEV               MISSING PA
422 METEOROLOGICAL ATTRIBUTE SIGNIFI               MISSING CODE TABLE 8005
423 LATITUDE (COARSE ACCURACY)                     MISSING DEG
424 LONGITUDE (COARSE ACCURACY)                    MISSING DEG
425 WIND SPEED AT 10 M                             MISSING M/S
```

Example BUFR decoder JSON output

```
{ "messages" : [
  [
    {
      "key" : "bufrHeaderCentre",
      "value" : 98
    },
    {
      "key" : "bufrHeaderSubCentre",
      "value" : 0
    },
    {
      "key" : "dataCategory",
      "value" : 7
    }, ...
```

Example BUFR decoder JSON output

```
{ "messages" : [
  [
    { ... }, ...
    {
      "key" : "unexpandedDescriptors",
      "value" :
      [
        1033, 1034, 1032, 1025, 1027, 1090, 1091, 1092, 301011, 301012,
        8005, 5002, 6002, 8005, 5002, 6002, 10051, 8005, 301023, 11012,
        108000, 31001, 8021, 4024, 8005, 301023, 10051, 8005, 301023, 11012
      ]
    }, ...
```

Example BUFR decoder JSON output

```
{ "messages" : [
  [
    { ... }, ...
    {
      "key" : "unexpandedDescriptors",
      "value" :
      [
        1033, 1034, 1032, 1025, 1027, 1090, 1091, 1092, 301011, 301012,
        8005, 5002, 6002, 8005, 5002, 6002, 10051, 8005, 301023, 11012,
        108000, 31001, 8021, 4024, 8005, 301023, 10051, 8005, 301023, 11012
      ]
    }, ...
    [
      [
        [ { ... },
          [ { ... },
            [ { ... },
              [ { ... },
                // and this goes on a few more levels...
```

Example BUFR decoder JSON output

```
{ "messages" : [
  [
    [
      [
        [ { ... },
          [ { ... },
            [ { ... },
              [ { ... },
                // and this goes on a few more levels...
```

There was really little or no documentation to make reading this JSON workable. The tools had **`man`** pages that didn't document the nested JSON keys or the syntax for using their API to select specific elements by name or position from within arrays of arrays of arrays of arrays...

Help!

I asked for help.

I was told,

"*Write a note and it will be forwarded to the right person.*"

I submitted my request.

I got an answer a couple of months later.

People are busy.

Meanwhile, in a week, I wrote a Racket program to solve my immediate problem.

Humorous aside

ECMWF is a wonderful organization, at the same time this license cracks me up every time I read it, which is every time I look at a source file.

> In applying this licence, ECMWF does not waive the privileges and immunities granted to it by virtue of its status as an intergovernmental organisation nor does it submit to any jurisdiction.

Immediate Solution

The JSON output was insanely nested and underdocumented.

The text output, while huge (100 times the size of the input), was at least comprehensible.

So I determined the patterns and wrote a **finite state machine** in Racket to read reliably that output, then pick out just the values I needed, and emit my own JSON output that was usable by downstream processes.

JSON others could use

```
{"storms":[
            {"storm":
             {"common":
              {"time":"2015-10-14 00:00:00",
               "name":"NORA",
               "basin":"E",
               "depression":18
              }
             },
             "positions":[
                          {"position":
                           {"hour":6,
                            "latitude":15.3,
                            "longitude":-150.5,
                            // other time-varying parameters of interest ...
                            }, { ... }
                          }
                         ]
             }
            ]
}
```

Then I had an idea

If I ever need to go into one of these files again, I want more flexible tooling.

That was the motivation for this side project.

The existing Fortran and C tools are very well built, and have been vetted for decades.

But they are not flexible enough for me.

Today people say: "homoiconic"

I want my program and the data to look the same.

I want my programs to reflect the shape of the data.
Gee, where have I heard that before?

Parentheses make for easy easy grouping.
With proper labeling, of course.

BUFR files have macro data expansions and templating.
Gee, I think I know a language that does that better than anything else.

You guessed it!

Racket!

BUFR Internals

- Six Sections
  - `0` : Indicator
  - `1` : Identification
  - `2` : Optional
  - `3` : Data Description
  - `4` : Data
  - `5` : `zzzz` (The End)

BUFR Straightforward Internals

- Lookup Tables
  - Many many entries
  - In layers of conditional lookup tables
  - Lets you build up to 4-dimensional data structures
- How lookup tables and data are interpreted is *very* context-dependent

BUFR Wonderings

You've probably looked at code in the past and seen it reimplemented as a bunch of lookup tables.

BUFR Wonderings

You've probably looked at code in the past and seen it reimplemented as a bunch of lookup tables.

When you look at BUFR, you start to think the other way around:

Can I convert these tables into functions?

It could make BUFR easier to use.

A few more things about BUFR

- At the time is was created (1988), it was considered advanced and flexible.
- Over time, limitations became apparent.
  - like only having years `00..99`
  - like only allowing 2^k-1 kinds of things, for small k.
- It has taken years of work by committees to keep it stable.

BUFR describes many kinds of weather data

- 7650 descriptors
  - 5152 unique descriptors
- more than 66 ways to describe time
- more than 51 ways to describe wind

BUFR descriptors, obvious

- Temperature and humidity data
- square metres per second
- 3-hour pressure change

BUFR descriptors, not hard

- Accuracy of the position of the centre of the tropical cyclone
- Clouds from waterfalls
- Descending orbit
- Funnel cloud touching surface
- Iridium communications, sampling on down transit
- Ship wind data
- Type of gas used in balloon

BUFR descriptors, Hmm...

- ASCAT satellite orbit and attitude quality
- Hoppers (nymphs, larvae), stage 1
- Instrument attached to animals other than marine mammals
- Large swarm or isolated adults, seen singly, duration of passage less than 1 hour ago
- Mutual inconsistency
- No countermeasures
- Prophylaxis

BUFR descriptors, Hmm...

- Repeat next 1 descriptor 87 times
- Rice - Evapotranspiration
- Spodoptera exempt
- Surface ozone bubler
- Tipping bucket method
- Violent
- Wrong
- parsec
    - ~3.26 light years (3.086×10^13 km), distance at which the mean radius of the earth's orbit subtends an angle of one second of arc

BUFR descriptors, Don't Laugh

- 11-20 icebergs, with growlers and bergy bits
- Balloon burst
- Not earth located because of bad time
- Reconstructed chirp used

BUFR descriptors, Esoteric

- (Z (Fl), Z (F2)) to attenuation to R
- Anomaly in Ultra Stable Oscillator Processing (USOP) value detected
- NASA Spectro-radiometer SAGE III Stratospheric Aerosol and Gas Experiment-III
- Sippican Deep Blue eqn: a=6.472, b=-2.16 (depth = a × time + 1.e-3 × b × time^2)

BUFR file challenges

- Data macro expansions and templating
  - They make it easier to pack more data in with less work.
  - They make it harder to figure out what's in a file, because you have more indirection.
- Get one bit wrong, and everything becomes nonsense.
- Lots of "special cases" added in over the years.

More Motivation

Existing *compiled* tools make it perfectly possible to decode most BUFR files.

But then you will have to wade through gobs of output to see what's there.

There is some interactivity built into a recently developed Python interface. But to use it, you really need to know what's inside to begin with.

I want! I want!

I want to ask general questions of BUFR files, so when someone asks me, *Can you tell if XYZ are in these files?* it's an easy answer, not a day of wading through volumninous output and looking things up. If you remember **`gdb`**, I want **`gdb`** for BUFR files.

I want to operate on collections of BUFR files: **`filter`** **`map`** **`fold`** **`reduce`**`...`, tree operations..., I want **`λ`** !

I want tooling that lets me poke around and maybe rearrange or edit things, especially if I want to edit or remove a section that was not properly defined, in case the BUFR file needs to be handled by someone else's compiled program that will **crash** because of something in the BUFR file.

You can see why Racket appeals to me

- Racket can consume and produce JSON, important for interfaces
- Racket has a REPL
- I can add in new code while I explore
- Racket has macros
- I can make any DSL for whatever a user exploring a BUFR file wants to type

Racket macros are so wonderful

- I can use macros to do similar things to BUFR's data macros and templates
- I can use macros to make reading and mirroring BUFR's data definitions a piece of cake
  - When WMO changes specs, my code can read member organizations' code and adapt

Wait, there's more!

- Racket can twiddle bits and handle streams
- Racket can handle many kinds of collections, sequences and iterators
- Racket has exception handling
- and contracts
- and types!

And more more!

- I can use the FFI to interface to existing vetted C and Fortran libraries
  - except when those libraries choke on a piece of BUFR that baffles them
- I can decode Fortran data structures directly
- I can pull those structures out and deal with them atomically
- I can use functional methods to keep state under control

I'm also excited because...

The weather community has a great many geeks. (I think of Gerry Sussman and realize I should say *nerds*.) There are large numbers of people out there in the world who just love to pour over weather files. But we all need more flexible and powerful tools!

These wonderful Weather Nerds probably don't know about Racket. I'd love for them to see what power awaits them, if they be worthy! I think this is an entirely new and **large** user community that could benefit from learning about Racket.

I want a weather project for public data to be licensed freely and flexibly, so people contribute and also get the most utility.

Where's the code?

- Still rapidly changing, but soon the code will go on public GitHub and in a few months will be an installable Racket package.
- But I have to figure out what to call this.
  - **`racket/weather ?`**
  - **`<something>/weather/bufr ?`**
- Naming is so hard... Suggestions welcome.
  - There will be other formats other than BUFR added in the future.
- Meanwhile contact info would be: **`geoff@knauth.org`** or **`@gknauth`**

Use your **Racket Mjolnir** to do good.

Thor and Loki visiting children in hospital