# Call by Meaning

Hesam Samimi

Yoshiki Ohshima

Alessandro Warth

SAP Labs

Chris Deaton

Cycorp

Todd Millstein

UCLA

Google Calendar | WorkFlowy - Organize you | www.hesam.us/callbymea | CbMJS Library | javascript string to date -

file:///Users/hesam/research/cdg-c/callbymeaning/cbmjslib/tests/zodiac1/index.html

Apps | G | P | G Cal | Wflowy | G | Y | G | News | A LA | Cal | sketchpad14 | cbmjslib tests | Nava tests | Cyc | SAP C | slack | CDG Cals | hesam | all@cdg

Code View

myApp

Your Birthdate: 1/4/6

Your Sign: Aries

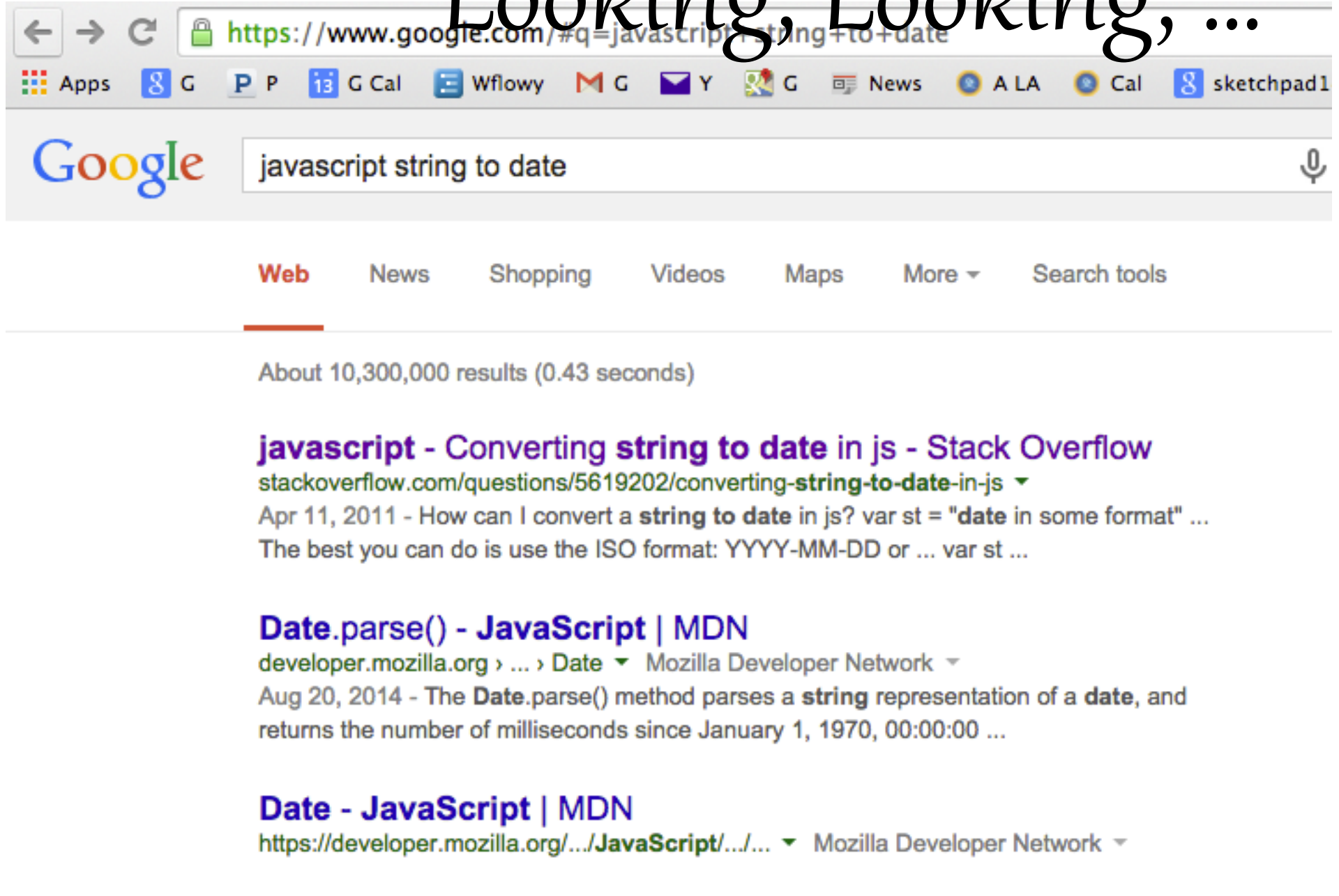
Code

```
run: function (userInput) {
    // parse the input
    this.userDateArray = this.getDate(userInput)
    if (this.userDateArray) {
        // convert into date object
        this.userDate = this.dateArrayToDate(this.userDateArray
        // compute zodiac sign
        this.userSign = this.getZodiacSign(this.userDate)
        canvas.getElementByName('signName').text = this.userSig
        // get a url to img of zoadiac sign
        this.signImgUrl = this.getZodiacSignImg(this.userSign)
        // get a uri to img of a CS person with that sign
        this.csPersonImgUrl = this.getCSPersonImg(this.userSign
        canvas.getElementByName('signImg').acyclicSet('src', th
        canvas.getElementByName('csPersonImg').acyclicSet('src'
    }
}
```

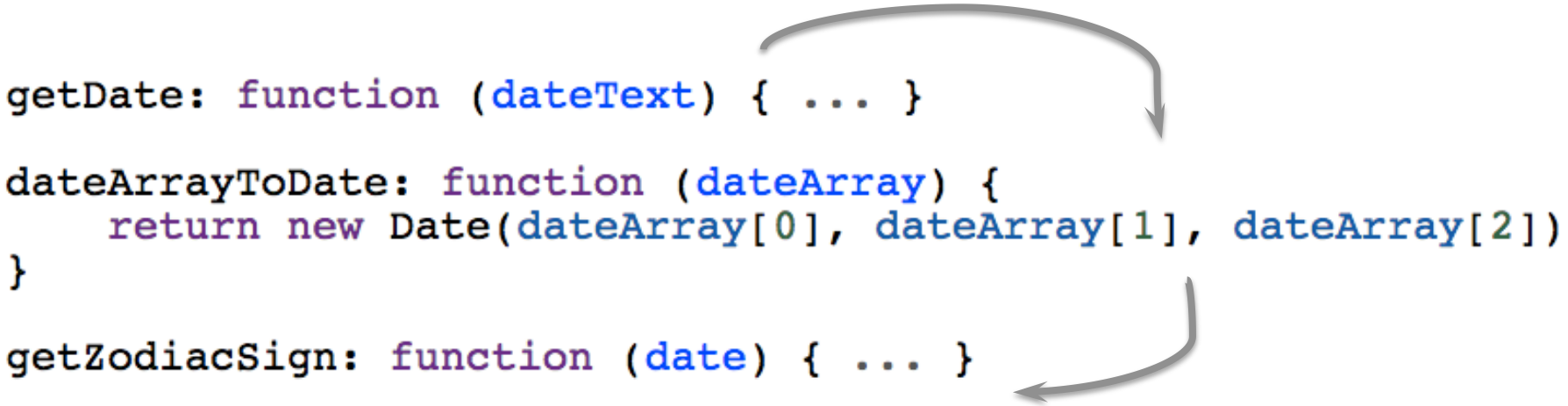
Values

```
name: "myApp"
userDateArray: [6,3,1]
userDate: "1906-04-01T07:00:00.000Z"
userSign: "Aries"
signImgUrl: "Aries.png"
csPersonImgUrl: "cs-wesley-clark.png"
```

# Looking, Looking, ...



# Hmmm, some glue code...



```
getDate: function (dateText) { ... }

dateArrayToDate: function (dateArray) {
    return new Date(dateArray[0], dateArray[1], dateArray[2])
}

getZodiacSign: function (date) { ... }
```

# Moral of Story

#1

We develop by composing existing software

# Moral of Story

#2

Lots of stuff

on the Internet!

# Moral of Story

#3

Finding

Understanding

Using

Composing

Maintaining

# ☹ Moral of Story

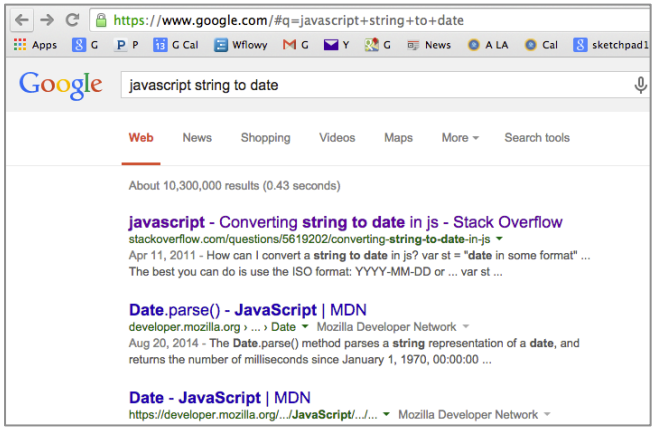
| | |
|---|---|
| Finding | manual! |
| Understanding | anti-automation! |
| Using | non-scalable! |
| Composing | reverse engineering! |
| Maintaining | hard-coding! |
| | fragility! |

# Status Quo in PL

## Finding

develop time (offline)

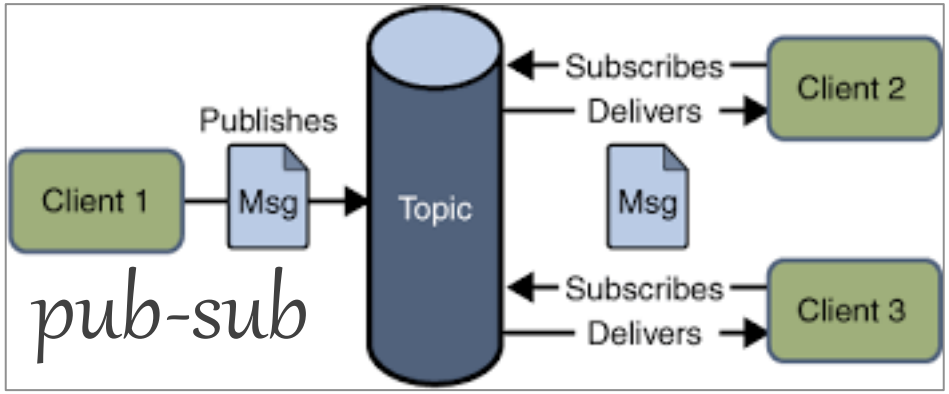
### Manual Search




dynamically (online)

### Access by Name

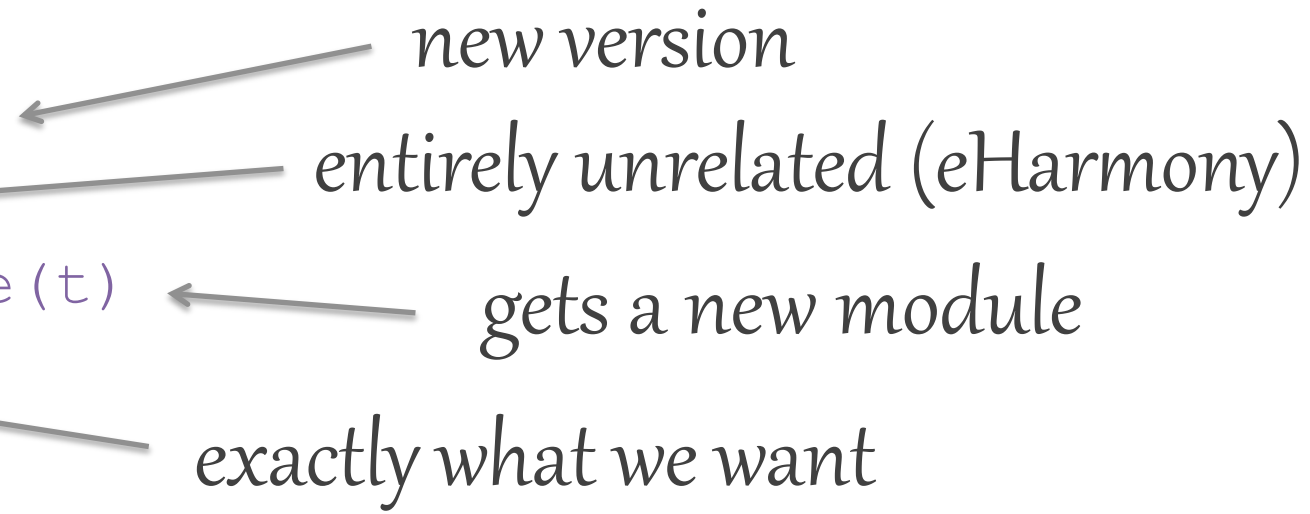
```
import Date
Date.getTime()
```



# Agreement on "name" doesn't scale!

```
getDate(t)

parseDate(t)
getDate(t)
Date.getDate(t)
foo(t)
```



# Agreement on "name" doesn't scale!

What's the alternative?

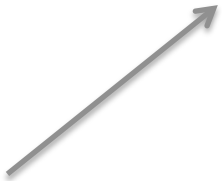
# Agreement on "name" doesn't scale!

## What's the alternative?

- type system?
- contracts?

# Your symbols mean nothing to me!

`com.calendar.Date`



just a name!

unrelatable to someone else

(meaning is in my head and associated code)

# What's the problem?

`com.calendar.Date`

lack of :

- grounding to the "real world"
- shared automatable semantic "understanding"

# How do we do it in the real world?

("It's on the tip of my tongue...")

Not a solution!

But a vision!

To set off a revolution!

of a PL where...

# "Call by Meaning" Vision

programming environment provides:

- semantic ontology & specification lang.
- grounding to the "real world"
- embedded vast shared "real world" KB
- reasoning engine (theorem prover)
- automatable semantic "understanding"

# Call by Meaning

*explain a component*

```
spec: function () {
  return "..."
}
```

# Call by Meaning

ask questions about a component

- discover API
- adapt its data

S Q

```
model = kb.ask(obj, "...")
```

repository

+

knowledgebase

+

prover

prove (S => Q)

dictionary binding logical vars in Q to values

# Call by Meaning

*discover a component*

S Q

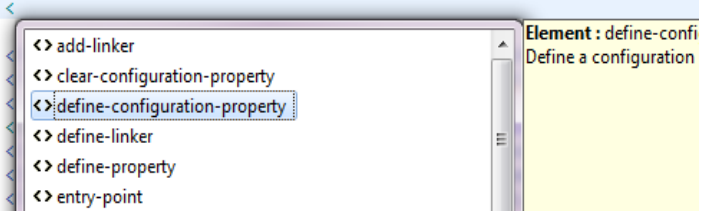
```
obj = kb.find("...")
```

*prove (S => Q)*

# Call by Meaning

## offline

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <!DOCTYPE module SYSTEM "http://google-web-toolkit.googlecode.com/svn/rel
3
4  <module rename-to='runpartnermaps' >
5      <!-- Inherit the core Web Toolkit stuff. -->
6      <inherits name='com.google.gwt.user.User' />
7      <
8
9
0
1
2
3
4
5
```




IDE Auto Suggestions

Part of Algorithm!


## online

Computers don't speak human or share human knowledge!

# 20+ Years of Semantic Web


ConceptNet 5

*Computers just got smarter.*

# Discovering

dateParser

```
getDate_spec: function () {
    return
    '(and (primaryFunction ?GETDATE SemanticParsing programUsed)'
      + '(programVariableIsaFunctionWithArg1 ?GETDATE ?Y CharacterString)'
      + '(programVariableIsaFunctionWithOutput ?GETDATE ?Z Date))'
}
```

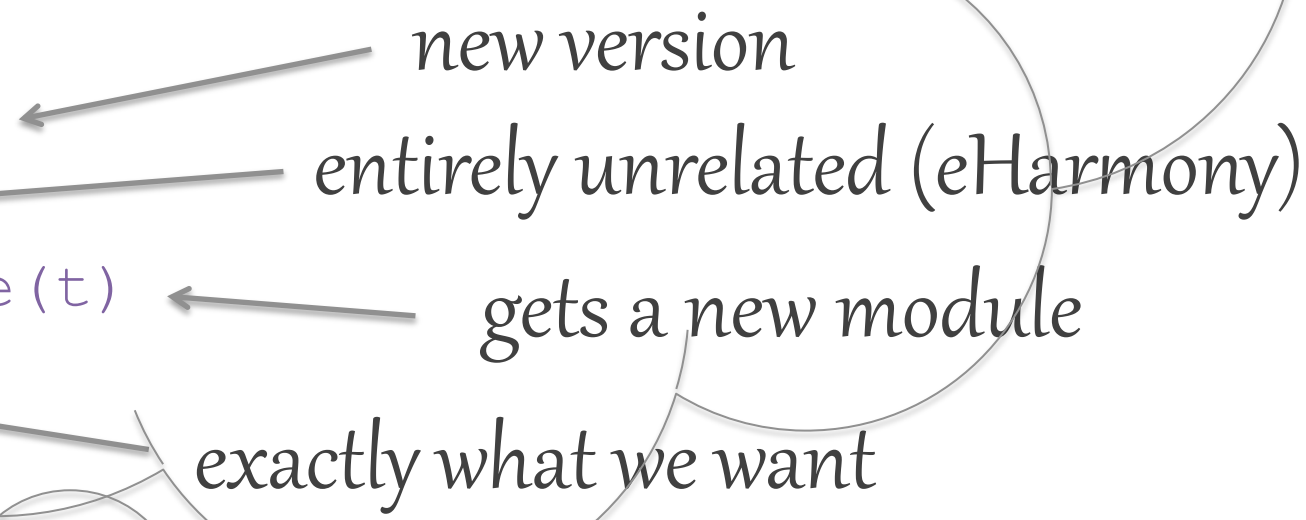
myApp

```
kb.findByMeaning(
    '(and (programVariableHasProperty ?THAT ?THAT-P)'
      + '(programParsesInputInto ?THAT-P CharacterString Date))',
    this, this.dateParserFound)
```

# Agreement on "name" doesn't scale!

```
getDate(t)

parseDate(t)
getDate(t)
Date.getDate(t)
foo(t)
```



# Discovering API

dateParser

```
getDate_spec: function () {
    return
    '(and (primaryFunction ?GETDATE SemanticParsing programUsed)'
       + '(programVariableIsaFunctionWithArg1 ?GETDATE ?Y CharacterString)'
       + '(programVariableIsaFunctionWithOutput ?GETDATE ?Z Date))'
}      + '(programVariableHasName ?GETDATE "getDate")'
```

myApp

{NAME: "getDate"}

```
kb.askByMeaning(
    '(and (programVariableHasProperty ?THAT ?THAT-P)'
       + '(behaviorCapable ?THAT-P SemanticParsing programUsed)'
       + '(programVariableIsaFunctionWithArg1 ?THAT-P ?THAT-Y CharacterString)'
       + '(programVariableIsaFunctionWithOutput ?THAT-P ?THAT-Z Date)'
       + '(programVariableHasName ?THAT-P ?NAME))',
    this, dateParser, false, this.onParserAPIDiscovered)
```

# Hmmm, some glue code...

```
getDate: function (dateText) { ... }

dateArrayToDate: function (dateArray) {
    return new Date(dateArray[0], dateArray[1], dateArray[2])
}

getZodiacSign: function (date) { ... }
```

dateParser

# Adapting

```
getDate_spec: function () {
    return  this.output ?
        '(programVariableIsaFunctionWithOutput ?GETDATE'
        + ' (DayFn ' + this.output[2]
        + ' (MonthFn ' + monthNames[this.output[1]]
        + ' (YearFn ' +  this.output[0] + '))) Date)'
        : undefined
}
```

4

myApp

{DAY: 4, …}

```
kb.askByMeaning(
    '(and (programVariableHasProperty ?THAT ?GETDATE)'
      + '(programVariableIsaFunctionWithOutput ?GETDATE ?DATE ?_)'
      + '(dayNumberOfDate ?DAY ?DATE)'
      + '(monthOfDate ?MONTH ?DATE)'
      + '(yearOfDate ?YEAR ?DATE))',
   this, this.dateParser, false, this.dateParserInfoReceived)
}
```

# Adapting+

dateParser

```
getDate_spec: function () {
    return  this.output ?
        '(programVariableIsaFunctionWithOutput ?GETDATE'
        + ' (DayFn ' + this.output[2]
        + ' (MonthFn ' + monthNames[this.output[1]]
        + ' (YearFn ' +  this.output[0] + '))) Date)'
        : undefined
}
```

myApp

{SIGN: "Cancer", …}

```
kb.askByMeaning(
    '(and (programVariableHasProperty ?THAT ?THAT-GETDATE)'
      + '(programVariableIsaFunctionWithOutput ?THAT-GETDATE ?DATE ?_)'
      + '(zodiacSignDateRange ?SIGN '
        + '(DayOfYearFn ?BEGIN-MONTH ?BEGIN-DAY) '
      + '(DayOfYearFn ?END-MONTH ?END-DAY))'
      + '(laterThanOrCooriginatingWith '
        + '(DayFn ?END-DAY (MonthFn ?END-MONTH (YearFn 1776))) ?DATE)'
      + '(laterThanOrCooriginatingWith ?DATE '
        + '(DayFn ?BEGIN-DAY (MonthFn ?BEGIN-MONTH (YearFn 1776)))))',
  this, this.dateParser, false, this.dateParserInfoReceived)
```

# Many challenges to making this a reality!

writing specs!

scalability of proving!

adopting a lingua-franca spec lang!

steep learning curve!

other forms of specs?

unpredictability/complexity!

safety & security!

# Related Work

- Semantic Web ← an "afterthought"
- PL approaches ← no grounding in "real world"
  - Zones project
  - Wolfram Language, Type Providers
  - API discovery (offline/online)

Thank you!

http://www.hesam.us/callbymeaning