Ftan

person:
Julien Jan
Axel
events: Max, Martin
Lukas
statistics: Dominik
Uwe, My-Tien
JAN
Axel

FtanML

reinventing markup

Michael Kay

- FtanML
    - The document markup language
- FtanGram
    - The schema language
- FtanSkrit
    - The scripting language

FtanML Example

```
<Polygon
   filled=true
   color="blue"
   corners=[[1,0], [1,1], [0,1], [0,0]]
   lineWidth=0.5
   label=|You are <i|so|> square!|>
```

FtanML Constructs

Elements

RichText

Numbers

Booleans

Strings

Lists

Null

Functions

Strings

"Max Weber"

'abcde\
fghijk'

"\xA0;"

'\r\n'

"He said \"I'm not\""

Escapes

- Special characters: \<, \\, \", …
- Whitespace: \s, \n, \t, \r, \S (=nbsp)
- Ignored whitespace: __________
- Hex Unicode codepoints: \x13a0f;
- Cells: \[✦qwertyuiop✦]

Lists

[ ]

["red", "green", "blue"]

[3,null]

[[1,0], [1,1]]

[<a>,<b>,<c>]

["red", 1, true]

Elements

- Optional name (any string)
- Zero or more attributes
  - Name (any string)
  - Value (any value)
- Optional content
  - Any value

Element examples

```
<>
<a>    <`*`>    <div [<p><p>]>
<`AT&T`=23.5>
<x=3 y=4 z=null>
<title |this is FtanML|>
<from=[0,1] to=[1,1]>
```

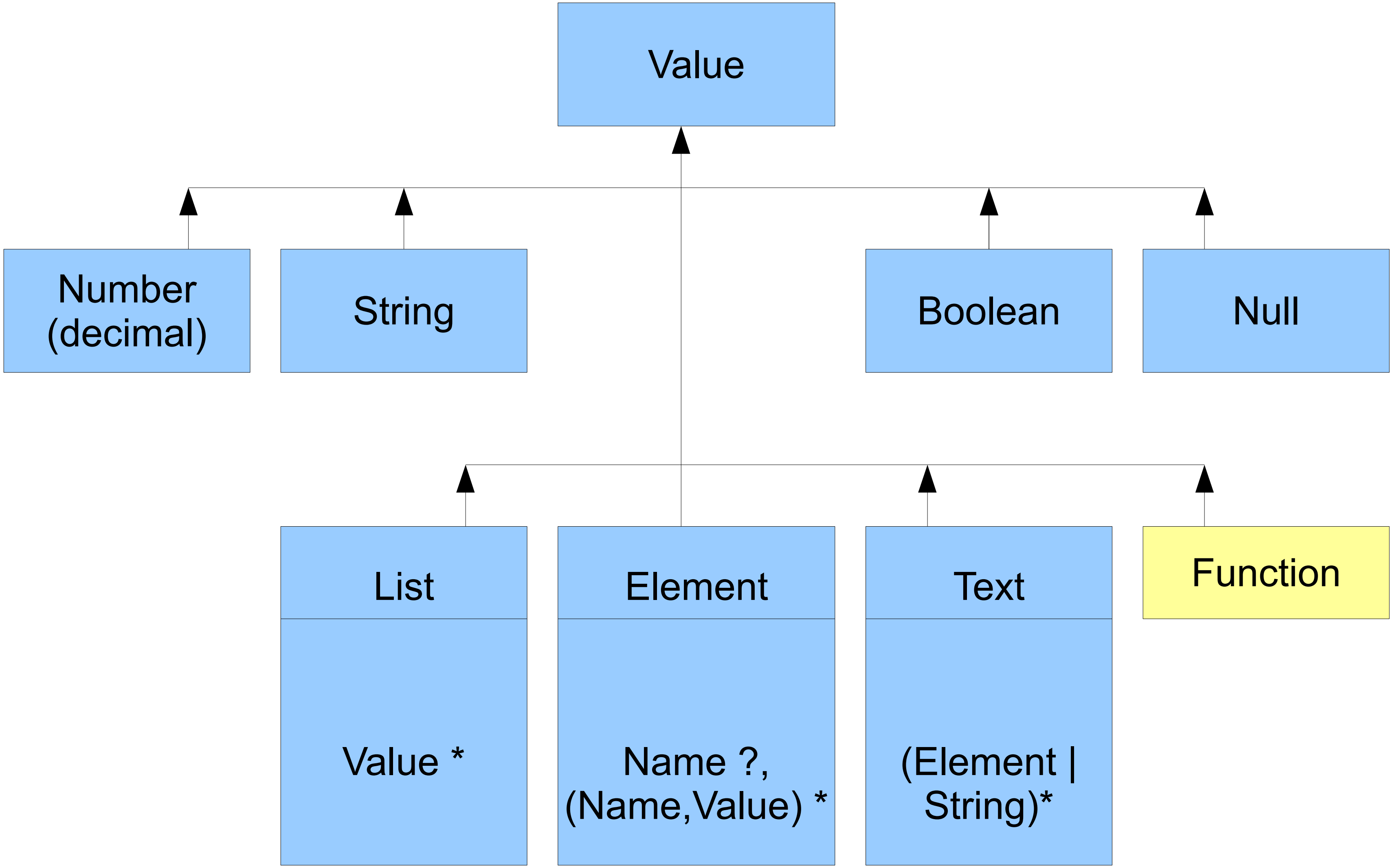
Namespaces

THERE ARE NO NAMESPACES.

Rich Text

```
<para |Drink lots of \
    <chem |H<sub 2>O|>\
|>
```

Data Model



Value
Number (decimal)
String
Boolean
Null
List
Value *
Element
Name ?, (Name,Value) *
Text
(Element | String)*
Function

What about nodes?

There are no nodes.

Values have no identity (only equality).

There are no parent pointers.

Whitespace?

Significant in strings and text.

Insignificant everywhere else.

FtanGram

The schema language for FtanML

> To grasp this sorry Scheme of Things entire, and ...
> re-mould it nearer to the Heart's Desire

FtanGram Types

- A schema is a set of (named) types
- A type has a FtanML representation as an element
- A type is a predicate
- There is no overt type hierarchy

Some types

<number ge=0 le=100 step=0.01>

<string pattern="[A-Z][0-9]">

<enum=["a", "b", "c", "d"]>

<assert={$@end > $@start}>

Composing Types

<allOf [<t1>, <t2>]>

<anyOf [<t1>, <t2>]>

<not <t1>>

<nullable <t1>>

Grammars

```
    <list grammar=
       <number occurs=[1,]>>

<list grammar=
   <seq [<number>,
          <string>,
          <boolean occurs=[1,5]>]>>
```

Particles

- A grammar is a particle
- A particle is an occurrence indicator plus one of:
  - <seq> + a list of particles
  - <choice> + a list of particles
  - a type

Element Proformas

```
<element form=
  <e x=<number>
     y=<number>
     z=<nullable<number>>
  >
>
```

Schema

```
<schema
  percentage = <number ge=0 le=100>
  zipcode = <string pattern="\[=\d{5}=]">
  emps = <list grammar=
                <emp occurs=[0,]>>
  emp = <element form=
         <emp first=<string>
               last=<string>
               bonus=<percentage>>
         >
>
```

FtanSkrit

The scripting language for FtanML

Variables

```
let x=3; let y=[4,5]; x+y[1]
```

Functions

```
let add={$1+$2};
add(2, 2)

let x=5;
let up={$+x};
7.up()
```

Filter/Select

```
let a = 1..100;
a?{$.mod(2) = 0}
```

```
emps?{$@salary > 50000}
```

```
let married = {$@status='M'};
emps?married
```

Map/Apply

```
let a = 1..100;
a!{$*2}
```

```
max(emps!{$@salary})
```

```
let age = {today().yearsSince($@dob)};
avg(emps!age)
```

Operations on Types

a.isA(<percentage>)

a.as(<percentage>)

Operations on Lists

```
let a = 1 ~+ 2..5 ++ [6,7,8,9] +~ 10
a = [1,2,3,4,5,6,7,8,9,10]
```

```
a[0] = 1
```

```
count(a) = 10
```

Operations on Elements

```
let n = 3;
let a = <e x=(n+1) y=(n+2)>;
a@x
```

```
<e>.add("x",2).add("y",3)
```

Implementation

https://github.com/FtanML-WG/Scala-Parser

Questions?