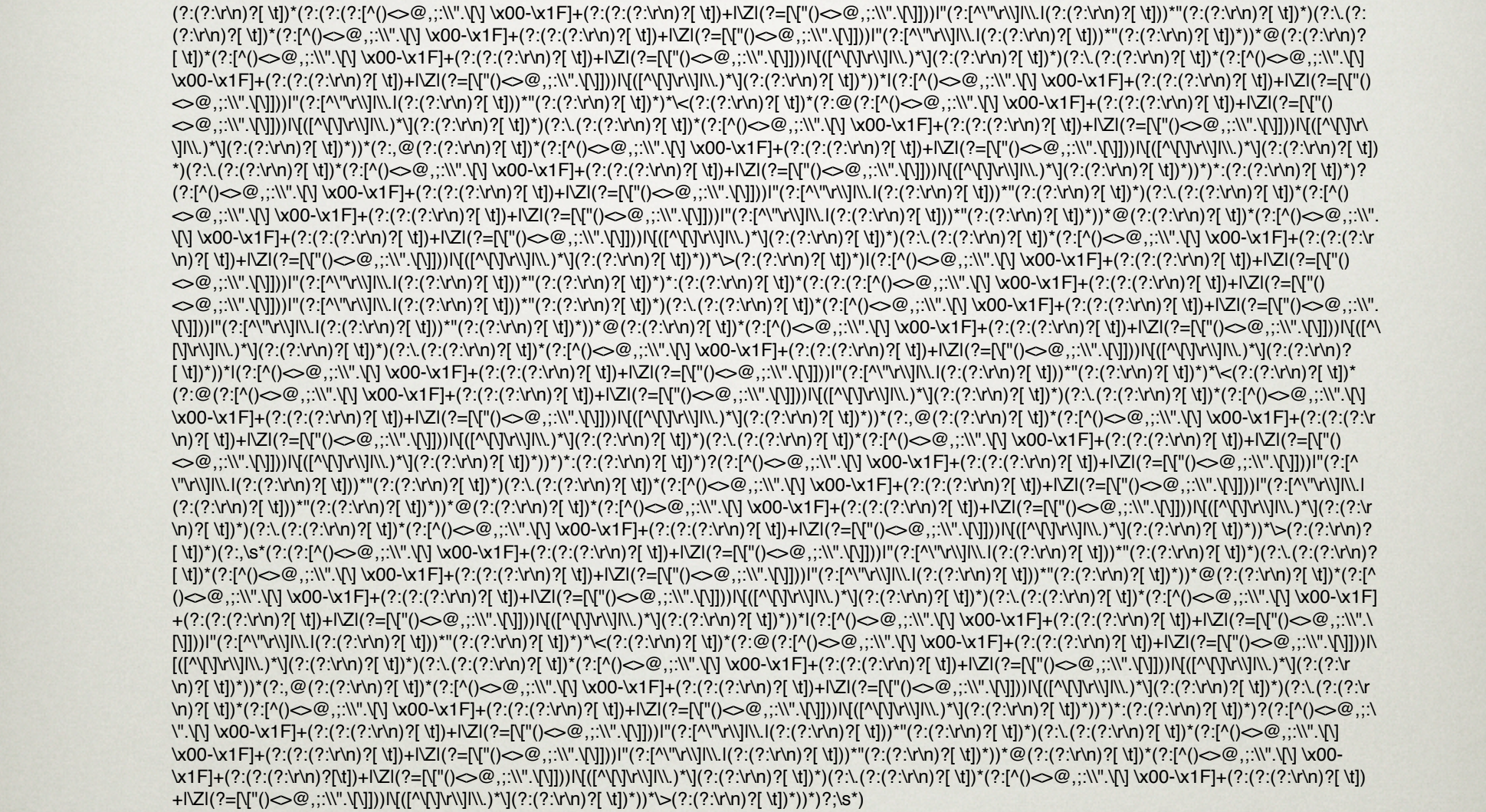
Implementing Domain Specific Languages

Alex Miller, BEA Systems

First, let's look at some examples.

^[A-Z0-9._%-]+@(?:[A-Z0-9-]+\.)+[A-Z]{2,4}$



```
SELECT COUNT(*)
FROM Albums JOIN Artists
WHERE Albums.ArtistID = Artists.ID
WHERE Artists.Name LIKE ‘G%’
```

/orders/order[id eq 162]

Cast on 90 sts.

Rows 1–17: K90.
Row 18: K8, P74, K8.
Row 19: K10, P16, (K2, P16) 3 times, K10.
Row 20: K8, P2, (K16, P2) 4 times, K8.
Row 21: K10, P16, (K2, P16) 3 times, K10.

1. e4 e5
2. Nf3 Nc6
3. Bb5 a6
4. BxN

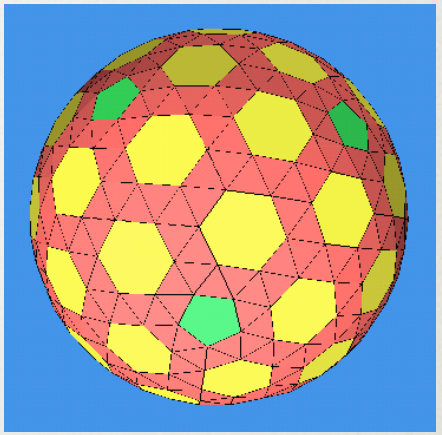
1. 4–2: 8/4, 6/4
2. 2–2: 13/9(2)
3. 3–1: 8/5, 6/5
4. 1–1: bar/23, 4/3*(2)
5. 4–3: bar/21, 24/21
6. 6–6: 24/18, 13/7, 9/3(2)

5–2: 24/22, 13/8
3–2: 24/22, 13/10
5–2: 22/17*/15
4–4: bar/21, 15/11, 6/2*(2)
6–3: 21/15, 13/10
13/7*, 8/7

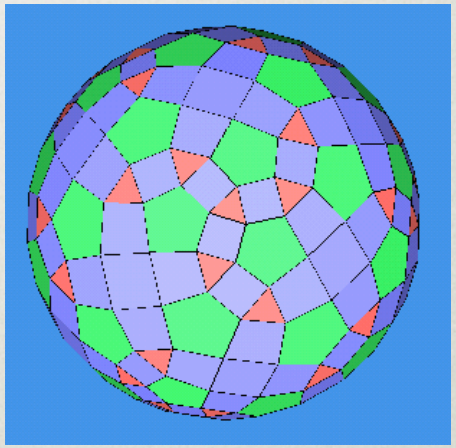
Conway notation for polyhedra

eptI

C
T
O
eptI
t10k10tD



sdk5sI

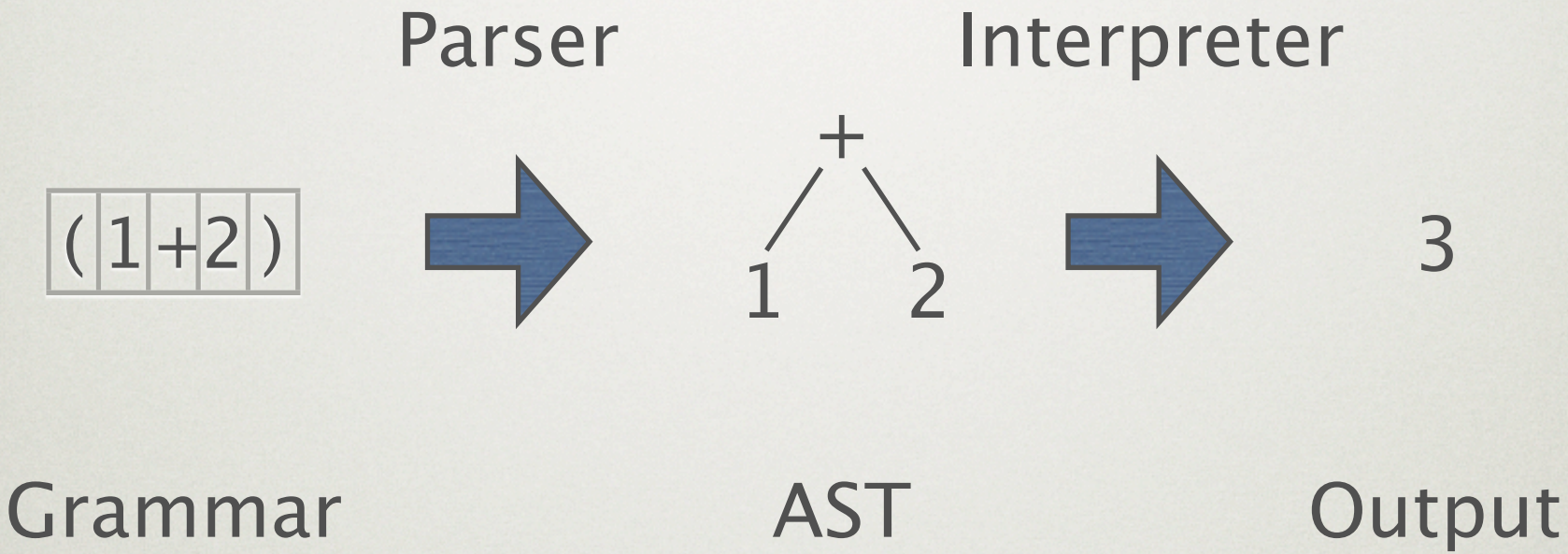


egH

How are these DSLs similar?

- Text-based
- Precise
- Concise
- Declarative (usually)
- Implicit context

What is the anatomy of a DSL?


Parser
Interpreter
(1 + 2)
+
1
2
3
Grammar
AST
Output

Internal vs External

Let’s look at some implementations

Fluent Interface

jMock Expectations

```
mock.expects(once())
  .method("receive").
  .with( eq(message) );
```

ProcessBuilder (java.lang in JDK 5)

```
Process p = new ProcessBuilder(“ls”, “-l”)
 .directory("~/stuff")
 .start();
```

BeanShell

- Syntax is superset of Java (so all GPL available)
- Still some noisy syntax (; , (), etc)
- Can use global variables and functions to save implicit context
- Can be extended to allow managing multiple contexts

XML

- Custom syntax
- Embedded in XML
- Parser built-in
- Examples abound...
  - J2EE deployment descriptors
  - Spring config
- Harder to read/write than custom DSLs, but easier to program

Antlr

- Full control over custom grammar
- Define grammar, generate
  - Lexer – translates stream of chars to stream of tokens
  - Parser – accepts sentences in grammar, create symbols
- More work than other approaches

When should I use a DSL?

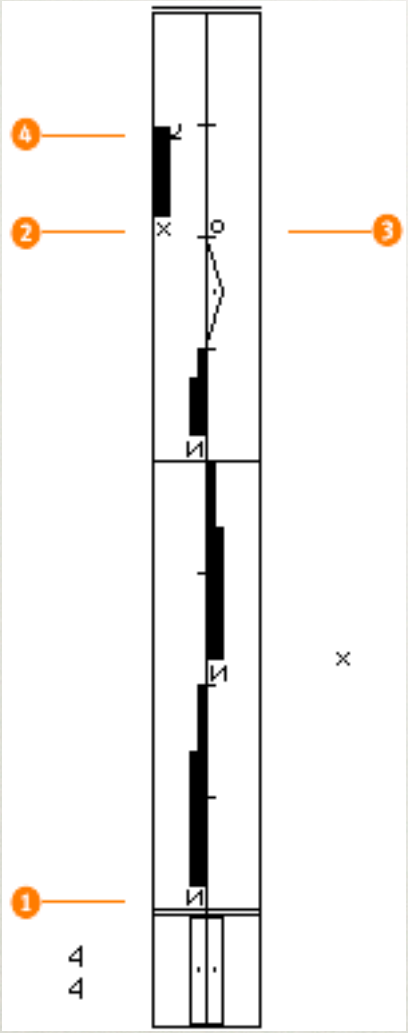
Pros

- Easy for domain users to understand
- Closer to domain than GPL version
- Concise due to implicit context
- Bring system closer to user
- Decouples users from implementation
- Portable

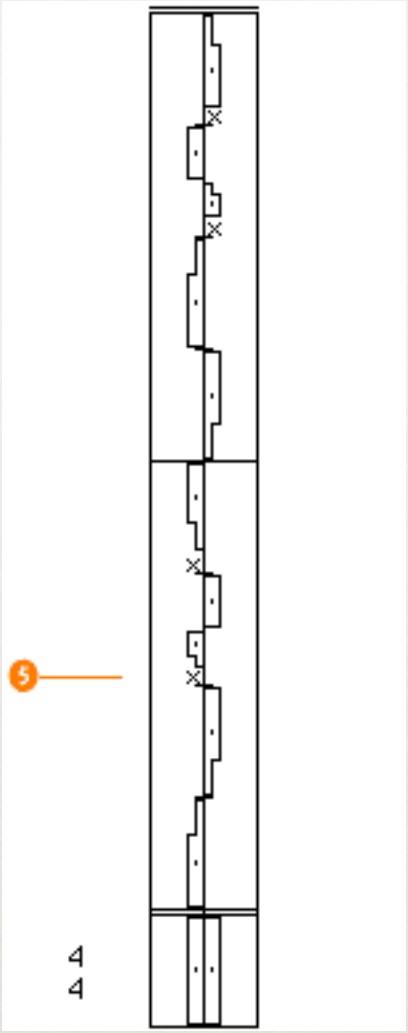
Cons

- Can be harder to design/implement
- No transfer from other languages
- Lack of general purpose constructs
  - Looping, variables, etc
- Hard to integrate multiple DSLs

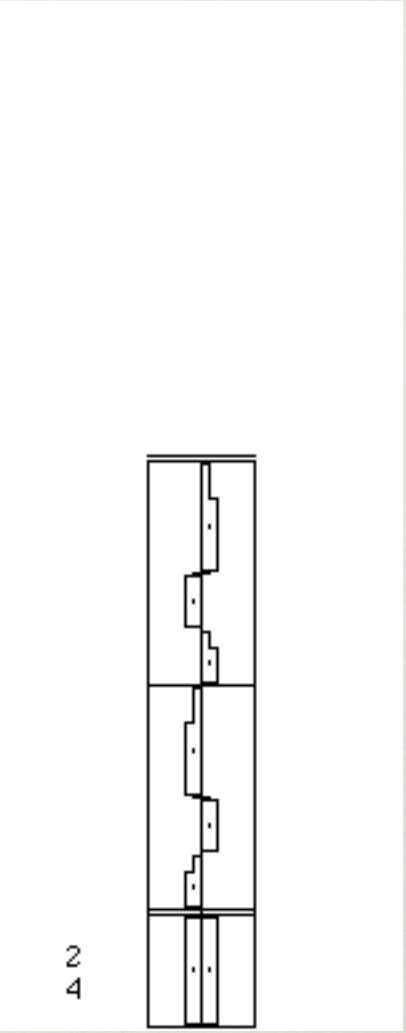
A visual DSL just for fun



Tango



Cha-cha



2-step

tech.puredanger.com