

Boa - Introduction

- Level 0 -

Overview

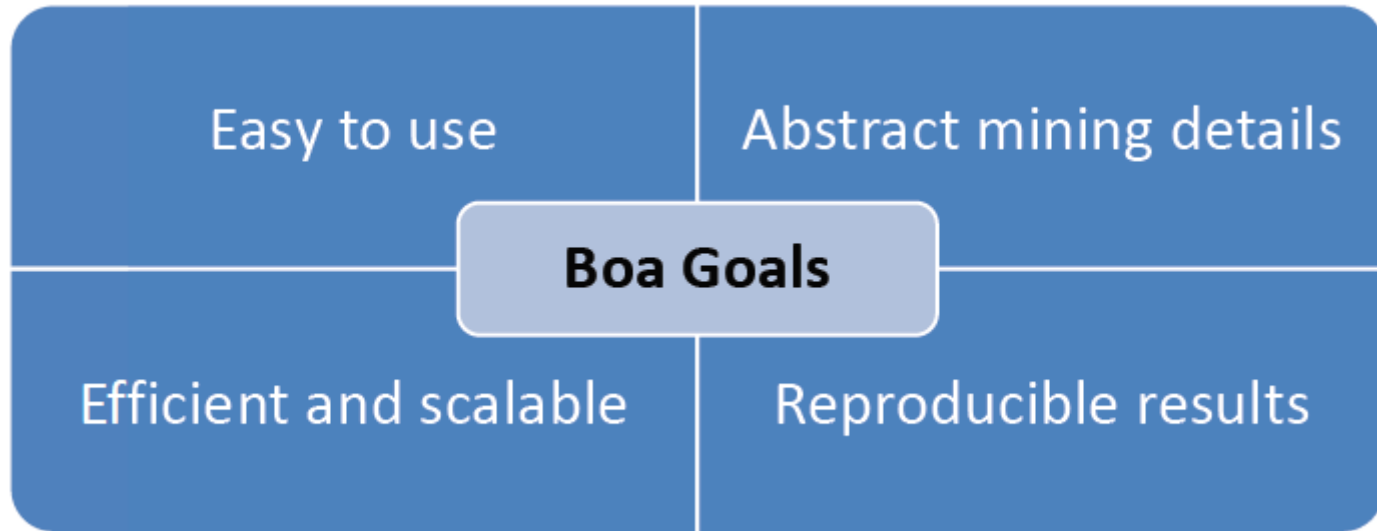
- What is Boa?
- Best uses for boa
- Boa: The Basics
- Syntax
- Examples
- Sources / Suggested Resources

What is Boa?

- Domain-specific language (DSL)
- Mines software repositories
- Software as a Service (SaaS)
- Has its own compiler
- Ex. Query Language for Git Repositories

The logo for Boa, consisting of the lowercase letters 'boa' in a stylized, rounded font. The letters are a dark red color with a yellow-to-orange gradient shadow or outline, giving it a 3D effect.

But why Boa?



Boa: The Basics

How to use Boa?

1. Login to <http://boa.cs.iastate.edu/boa/> .
2. Select Run Example.
3. Select an existing example or create your own query.
4. Magic!

Syntax of Boa language

- Declaring a variable: **name:Type = value;**
- Input variable
 - **p:Project = input;**
 - (this is used at the start of all boa programs)
- Output:
 - Note that Boa does not have explicit print statement; it prints out every output variable after execution.
 - **counts: output aggregator(param)[indices] of T [weight];**
 - Typically assigned values later on in the code.
 - “output” MUST be included in variable declaration. There are multiple output aggregators.
- “<<” operator:
 - Takes value and add it to the aggregator (aka. Adds value to variable)

Example 1

What are the ten most used programming languages?

```
1 # Counting the 10 most used programming languages
2 p: Project = input;
3 counts: output top(10) of string weight int;
4
5 foreach (i: int; def(p.programming_languages[i]))
6     counts << p.programming_languages[i] weight 1;
```

Example 1

What are the ten most used programming languages?

```
1 # Counting the 10 most used programming languages
2 p: Project = input;
3 counts: output top(10) of string weight int;
4
5 foreach (i: int; def(p.programming_languages[i]))
6     counts << p.programming_languages[i] weight 1;
```


Example 2

How many projects use more than one programming language?

```
1 # Counting the number of projects written in more than one languages
2 p: Project = input;
3 counts: output sum of int;
4
5 if (len(p.programming_languages) > 1)
6     counts << 1;
```

Example 2

How many projects use more than one programming language?

```
1 # Counting the number of projects written in more than one languages
2 p: Project = input;
3 counts: output sum of int;
4
5 if (len(p.programming_languages) > 1)
6     counts << 1;
```

Example 3

```
1 #Shows how many time each programming language is used.  
2 p: Project = input;  
3 counts: output sum[string] of int;  
4 foreach (i: int; def(p.programming_languages[i]))  
5     counts[p.programming_languages[i]] << 1;
```

Example 3

```
1 #Shows how many time each programming language is used.
2 p: Project = input;
3 counts: output sum[string] of int;
4 foreach (i: int; def(p.programming_languages[i]))
5     counts[p.programming_languages[i]] << 1;
```

Complex Example

- Normally queries aren't this long!
- We will work up to this level.

How is transient keyword used in Java?

```
1 # How is transient keyword used in Java?
2 p: Project = input;
3 TransientTotal: output sum of int;
4 TransientMax: output maximum(1) of string weight int;
5 TransientMin: output minimum(1) of string weight int;
6 TransientMean: output mean of int;
7
8 count := 0;
9 s: stack of int;
10
11 visit(p, visitor {
12   before node: CodeRepository -> {
13     # only look at the latest snapshot
14     # and only include Java files
15     snapshot := getsnapshot(node, "SOURCE_JAVA_JLS");
16     foreach (i: int; def(snapshot[i]))
17       visit(snapshot[i]);
18     stop;
19   }
20   before node: Declaration -> {
21     # only interested in fields, which only occur inside (anonymous) classes
22     if (node.kind == TypeKind.CLASS || node.kind == TypeKind.ANONYMOUS) {
23       # store old value
24       push(s, count);
25       count = 0;
26
27       # find uses and increment counter
28       foreach (i: int; def(node.fields[i]))
29         foreach (j: int; node.fields[i].modifiers[j].kind == ModifierKind.OTHER
30                 && node.fields[i].modifiers[j].other == "transient")
31           count++;
32     } else
33       stop;
34   }
35   after node: Declaration -> {
36     # output result
37     TransientTotal << count;
38     TransientMax << p.id weight count;
39     TransientMin << p.id weight count;
40     TransientMean << count;
41
42     # restore previous value
43     count = pop(s);
44   }
45 });
```

Sources / Suggested Resources

- ISU CS Website: <http://boa.cs.iastate.edu/docs/index.php>
- UCLA Website: <http://web.cs.ucla.edu/~shyoo1st/boa/>