The Craftsman: 50
Brown Bag VII
Ruby

Robert C. Martin
13 June, 2007

October 31, 1944

"We need to plan a mission to Clyde as soon as possible!" Von Braun said to the monthly meeting of the Contingent. The Contingent had maintained their identity as thought leaders during the whole tenure of the Nimbus project. New strategies and ideas often came from this now august body of idealists and risk-takers.

"But that will compromise our mission to Mars!" said Sergey Korolyov. "We do not have the time to waste! We must see what is on Mars so we can make plans to colonize it, or we will all vanish without a trace!"

Von Braun nodded. "Yes, Sergey, we must go to Mars. And you are right, this new mission will delay that goal slightly. But we must also find out whether it might be possible to use an Ulam engine to shove Clyde out of the way. If that is possible then…"

"It's not! You can't push on a pile of rubble. There's nothing to push. The pusher will just…"

"No Sergey, we think we can! Look." Von Braun held up a diagram of oddly shaped space vessel. "Notice that this vessel has two Ulam drives, one at each end. We call it a "Piston" for reasons that will become clear momentarily. We line a dozen, two dozen, a hundred of these pistons very close to Clyde, so that their posteriors point at him. Then we start both *engines on each piston. The ships don't move very much because both engines have equal and opposite thrust; but the posterior thrust of the ships pushes against Clyde!"*

Sergey, and everyone else, was stunned into silence. Von Braun saw their reaction and his composure crumbled. He hung his head, and began to sob. And as the sobs wracked through his body he managed to say: "We will push him away from us. We will push him away from us."

Tuesday, 5 Mar 2002, 1115

We had to change to a larger conference room because so many people showed up for today's session. There were people there I had not seen since school. There were even some people who's names I wasn't sure of. I counted 25 people total.

I saw Avery sullenly file in right behind Jean. I would have thought he'd want to stay as far away from her as possible; but maybe she had other ideas. I made a mental note to ask him the next time we had a moment to ourselves.

Jasper had claimed the front of the room and was pacing back and forth with his big grin beaming at the attendees as they settled themselves in. I tried not to let my apprehension show as I braced myself for his "over the top" style.

"Ruby!" He suddenly cried. "Ruby, Ruby, Ruby!"

I saw a number of the journeymen shift uncomfortably in their seats. Jasmine let out a frustrated

groan.

Jasper beamed and then rolled his eyes at the crowd. "Yes, I know, you've heard it before."

"Yeah, we have Jasper." Jasmine voice dripped with fatigued derision. "Those guys on the *Ulam* must have found some pretty potent weeds on Altair IV."

There were high-fives, sniggers and giggles amongst the journeymen. The rivalry between the *Dyson* and the *Ulam* was a matter of legend and pride. But Jasper would not be shushed. Instead he gushed:

"Yes, those oh-twenty guys have done some pretty strange things in the past, but you have to admit that their recent software productivity is giving those of us at sixteen-sixty-under a run for our money. I mean this new *Rails* thing is helping them get stuff done at a fee-ROHSHUS pace."

"It's not safe!" Jasmine asserted to the murmured agreement of some others. "They're going to have run-time errors!"

Jasper gave his head an exaggerated shake. "NOooo, their numbers say different! Remember we learned TDD from *them* in '99 They've been unit testing for over eight years, and their runtime error rate is well below *ours*!"

Jasmine green eyes flashed with self-assurance, but she stammered in frustration. "B-But giving up compile time type safety is just – nuts!" There were nods and grunts of agreement from the others. Jasper spread his arms in appeal as a half-dozen side conversations broke out. It looked like Jasper's talk was over before it began.

I was pretty confused. I didn't know much about the Ruby language, just that some fellow named Matz on the *Ulam* had invented it some years back. I was about to raise my hand and ask Jasper to explain when Jean stood up and looked around the room expectantly. I could see Avery glowering behind her, keeping his eyes on the floor. All the side conversations quickly died down.

"My dears, my DEARs! Yes, it's true that we don't necessarily agree with everything they do on the *Ulam*. Goodness knows, I've had my own misgivings about all the shenanigans they are up to. But we have to remember that those brave folks have been to almost as many planets as we have, and remain among the survivors of our fleet."

This indirect reminder of the losses to our fleet took some of the starch out of the more vocal objectors. Jasmine dropped her shoulders, sighed, and took Jerry's hand. Jean continued:

"Now let's all do Jasper the honor of listening to what he has to say. I'm sure there is something we can all learn from him today."

Jean sat down and whispered something in Avery's ear. He grimaced and then brought his eyes off the floor and towards the front.

"SOooo, AS I was SAYing:" Jasper grabbed the floor back from Jean and put every ounce of ham into it. "Ruby! Remember Adelade's great talk yesterday? She told us about using the Abstract Factory pattern to break compile time dependencies in the SMCRemote project. She wanted to create a plug-in architecture that would allow other code generators to be added without affecting the core system. Well, here's how you'd solve that problem in Ruby."

Jasper told the wall to watch his fingers, and he began typing in mid-air. The following code appeared on the wall.

```
require 'cpp_generator'
require 'java_generator'

class GeneratorFactory
  def createCpp
    return CppGenerator.new
  end

  def createJava
    return JavaGenerator.new
  end
end
```

"That really doesn't look all that interesting Jasper." Jasmine said. "It's not much different than that Java version that Adelade showed us yesterday".

"You are almost right Jazzy." Jasper said with a toothy grin. I could see Jasmine redden with anger. "Notice that there's no base class. Adelade's Java factory implemented an interface that had the two `create` methods."

Jerry blocked Jasmine's angry response by saying "OK, Jasper, but that just means that you've lost the plug-in structure. The `SMCRemote` program is going to depend upon `GeneratorFactory`, and will therefore know about `CppGenerator` and `JavaGenerator`."

"Nosireee, Jerry-O! That's the whole point! You see, `SCMRemote` has no idea what kind of object the factory is.

"Huh? That makes no sense. It's got to know otherwise it wouldn't be able to call `createCpp` or `createJava`."

Jasper was clearly on the verge of an ebullient ejaculation but Jasmine overrode him. "No, Jerry, he's right. In Ruby types are checked at runtime, not at compile time, so the `SMCRemote` program does not need to know what type of object the factory is."

Jasmine strode to the wall and all but shoved Jasper out of the way. She wrote the following code.

```
class SMCRemote
  def initialize(factory)
    @factory = factory
  end

  def selectLanguage(language)
    case language
      when "C++"
        @generator = @factory.createCpp
      when "Java"
        @generator = @factory.createJava
    end
  end

  def generator
    @generator
  end
end
```

"Notice," She said, "that there are no require statements. SMCRemote does not need to know anything at all about the factory. It simply assumes that the createCpp and createJava methods exist. If they don't, then there will be a runtime error. And THAT is what I mean about how unsafe this language is. The compiler won't tell you if a method doesn't exist. You have to wait until…"

Jasper finally got back into the act and interrupted Jasmine. "Right, Jazzy, we've heard it before. The point is that the plugin structure survives. `SMCRemote` does not need to know anything at all about `GeneratorFactory`."

Thouroughly confused, I stood up and asked some questions. "I'm sorry, this is probably basic, but I'm not sure I understand all that syntax. I presume `def` is used to define a function?"

"You got it Fonze!"

"And the @ sign is somehow connected with instance variables?"

"Yep, right on! Any variable prefixed with an @ sign is an instance variable."

"OK, so you don't have to declare instance variables anywhere?"

"Nope, no declarations at all! Isn't that cool!"

"I don't know how cool this is, could I see it work?"

Jasper gave me one of the widest and toothiest smiles I had ever seen. It nearly split his head in two. "Why sure! Just look at this coolness!" And he typed the following on the wall.

```
describe SMCRemote do
  before do
    @factory = GeneratorFactory.new
    @smc = SMCRemote.new(@factory);
  end

  it "should create a JavaGenerator when passed 'java'" do
    @smc.selectLanguage("Java")
    @smc.generator.class.name.should == "JavaGenerator"
  end

  it "should create a CppGenreator when pass 'C++'" do
    @smc.selectLanguage("C++")
    @smc.generator.class.name.should == "CppGenerator"
  end
end
```

This syntax was totally foreign to me. Still, after a minute I could see that this was some kind of strange unit test. The `before` function was like `setup` in JUnit. The `it` blocks were like test functions. The `should` verbs were similar to asserts. It all made an odd kind of sense.

Jasper looked at us all expectantly, and then with a flourish he ran these tests, and they passed.

"I…Just…Don't…Get…It." Jerry cried. "I don't see how `SMCRemote` can call either of those `create` functions without knowing the type of `@factory`. It doesn't make any sense."

"Think of it this way, Jerry." Jasmine said. She had been very patient with Jerry lately. "`SMCRemote` doesn't *know* that it can call `createJava` on the `@factory`, so it asks `@factory` if it has a method named `createJava`. If so, it calls it. Otherwise it bombs."

Jerry was incredulous. "You mean it passes `createJava` as a string, and does a search? That would be really slow."

"And dangerous." She replied while nodding.

"That's not the reason that Ruby is slow." Jasper blurted.

"But it is pretty slow, isn't it." Jasmine quipped.

"It's getting faster." He wheedled.

People started to file out of the room. Our time was long passed.

"I don't know about this goofy language." Jerry said to Jasmine.

"I do…" I heard her say as they left.

Jean and Avery walked out. "Back to work now my dears. Goodness! We are quite late."

Soon it was just Jasper and I. Jasper stood motionless, staring at the door with a puzzled look on his face. Eventually he looked at me and mumbled. "Well, that certainly didn't go as well as I thought it would."

I ignored that and said. "Jasper, there's a lot I don't understand about Ruby, but I did notice once thing. You were able to keep the plug-in structure, the complete isolation of `SMCRemote` from the `Generators`, without using any inheritance. "

Jasper looked at me out the corner of his eye, and I saw that face splitting smile begin to grow.