Understanding DRL

(Drools Rules Language)

Red Hat

Before we start

Drools community

- https://kie.zulipchat.com/ (**#drools** and **#kogito** channels)
- **Drools Usage** and **Drools Setup** mailing list
- https://groups.google.com/g/drools-usage
- https://groups.google.com/g/drools-setup
- https://**kogito**.kie.org/community/

Understanding DRL

(Drools Rules Language)

https://www.drools.org/learn/documentation.html

Declarative vs. Imperative

Imperative Programming

Declarative Programming

Creating a programming language for a rule system

Many ways to define business rules

- DRL
- Executable Models
- Decision Tables
- DMN Models

What is a business rule?

Anatomy of a rule

```
when: a condition is true
then: do something
```

Example Validation Java #1

"You need to insert your first name to sign up to this service"

```
public void validate(InputData inputData) {
      if(inputData.getFirstName() == null) {
          addValidationError(inputData.getUser(), "First name is required");
      }
  }
```

Example Validation Java #2

Multiple Validation

```
public void validate(InputData inputData) {
    if(inputData.getFirstName() == null) {
        addValidationError(inputData.getUser(), "First name is required");
    } else if (inputData.getLastName() == null) {
        addValidationError(inputData.getUser(), "Last name is required");
    }
    ... and many more
}
```

Example: Validation

"You need to insert your first name to sign up to this service"

```
rule "first name exists"
when
    InputData(firstName == null, $u : user)
then
    validation.error($u, "First name is required");
end
```

Example: Validation

Multiple validation

```
rule "first name exists"
when
    InputData(firstName == null, $user : user)
then
    validation.error($user, "First name is required");
end

rule "address exists"
when
    AddressData(address == null, $address : address)
then
    validation.error($address, "address is required");
end
```

Create a simple Drools project

Let's show some code

Session

- Stateful Session
- Stateless Session

Stateful Session

```
Customer c = new Customer();
c.setCategory(Category.Silver);

Order o1 = new Order(c, valueOf(100));

session.insert(c);
session.insert(o1);

session.fireAllRules();
assertEquals(Category.Silver, c.getCategory());

Order o2 = new Order(c, valueOf(200));
session.insert(o2);
session.fireAllRules();

assertEquals(Category.Gold, c.getCategory());
```

Stateful Session

Rule updating the customer

```
rule "Will be promoted to gold when they make an expensive order"
when
    $o : Order( amount >= 200, $c: customer)
then
    $c.setCategory(Category.Gold);
end
```

Update the customer without notifying the engine

```
rule "Will be promoted to gold customer when makes an expensive order"
when
    $c : Customer()
    $o : Order( amount >= 200, customer == $c)
then
    $c.setCategory(Category.Gold)
end

rule "User will receive an email when becomes a gold customer"
when
    $c : Customer(category == Category.Gold)
then
    System.out.println("Congratulation, you've become a gold customer");
end
```

Modify

```
rule "Will be promoted to gold customer when makes an expensive order"
when
    $c : Customer()
    $o : Order( amount >= 200, customer == $c)
then
    modify($c) {
        setCategory(Category.Gold);
    }
end

rule "User will receive an email when becomes a gold customer"
when
    $c : Customer(category == Category.Gold)
then
    System.out.println("Congratulation, " + $c + "you've become a gold customer");
end
```

Inserting facts in the DRL

Defining control types

```
declare IsGoldCustomer
    customer: Customer
end

rule "Classify Customer - Gold"
    when
        $c: Customer( category == Category.Gold )
    then
        insert(new IsGoldCustomer($c));
end
```

Inserting facts in the DRL

Defining control types

```
declare IsLowRangeItem
    item: Item
end

rule "Classify Item - Low price"
    when
        $i: Item(cost < 10.00)
    then
        insert(new IsLowRangeItem($i));
end

rule "Suggest gift"
    when
        IsGoldCustomer($c: customer)
        IsLowRangeItem($i: item)
    then
        System.out.println("Suggest giving a gift of item "+$i.getName()+" to customer "+$c.getName());
end
```

Deleting objects into working memory

```
rule "Delete expiring item"
    when
        $now : Date()
        $i : Item( expiringDate after $now)
    then
        delete($i)
end
```

Debugging the evalution of a rule

Using the DebugAgendaEventListener

The order of execution is not obvious, what do we do if we don't understand what's happening?

```
session.addEventListener(new DebugRuleRuntimeEventListener());
```

Debugging the evalution of a rule

An example of debug trace

```
15:04:16,797 INFO  ==>[ObjectInsertedEventImpl:
    getFactHandle()=[fact 0:1:274996233:274996233:1:DEFAULT:NON_TRAIT:org.example.Customer:Customer{category=Silver, name='customer'}],
    getObject()=Customer{category=Silver, name='customer'},
    getKnowledgeRuntime()=KieSession[0],
    getPropagationContext()=PhreakPropagationContext
        [entryPoint=EntryPoint::DEFAULT,
            factHandle=[fact 0:1:274996233:274996233:1:DEFAULT:NON_TRAIT:org.example.Customer:Customer{category=Silver, name='customer'}],
            originOffset=-1,
            propagationNumber=2,
            rule=null, type=INSERTION]
]

15:04:16,819 INFO  ==>[ObjectUpdatedEventImpl:
    getFactHandle()=[fact 0:1:274996233:274996233:5:DEFAULT:NON_TRAIT:org.example.Customer:org.example.Customer{category=Gold, name='customer'}],
    getObject()=org.example.Customer{category=Gold, name='customer'},
    getOldObject()=org.example.Customer{category=Silver, name='customer'},
    getKnowledgeRuntime()=KieSession[0],
    getPropagationContext()=PhreakPropagationContext
        [entryPoint=EntryPoint::DEFAULT,
            factHandle=[fact 0:1:274996233:274996233:5:DEFAULT:NON_TRAIT:org.example.Customer:org.example.Customer{category=Gold, name='customer'}],
            originOffset=-1,
            propagationNumber=6,
            rule=[Rule name=Will be promoted to gold customer when makes an expensive order, agendaGroup=MAIN, salience=0, no-loop=false], type=MODIFICATION]]
```

Stateless Session

```
StatelessKieSession statelessKieSession = kContainer.newStatelessKieSession("rules.simple.sl.discount");

Assert.assertNotNull(statelessKieSession);

Customer customer = new Customer();
customer.setCategory(Customer.Category.SILVER);

Order order = new Order();
order.setCustomer(customer);

Command newInsertOrder = ks.getCommands().newInsert(order, "orderOut");
Command newInsertCustomer = ks.getCommands().newInsert(customer);
Command newFireAllRules = ks.getCommands().newFireAllRules("outFired");
List<Command> cmds = new ArrayList<Command>();
cmds.add(newInsertOrder);
cmds.add(newInsertCustomer);
cmds.add(newFireAllRules);
ExecutionResults execResults = statelessKieSession.execute(ks.getCommands().newBatchExecution(cmds));
```

Executable Model

Another language to declare rules

Executable Model

Purpose

- To have faster startup time
- To provide an easier way to generate rules.
- Enable the cloud native support (Kogito - https://kogito.kie.org)

Executable Model

Benchmarks

`BuildKieBaseFromContainerBenchmark.getKieBaseFromContainer`

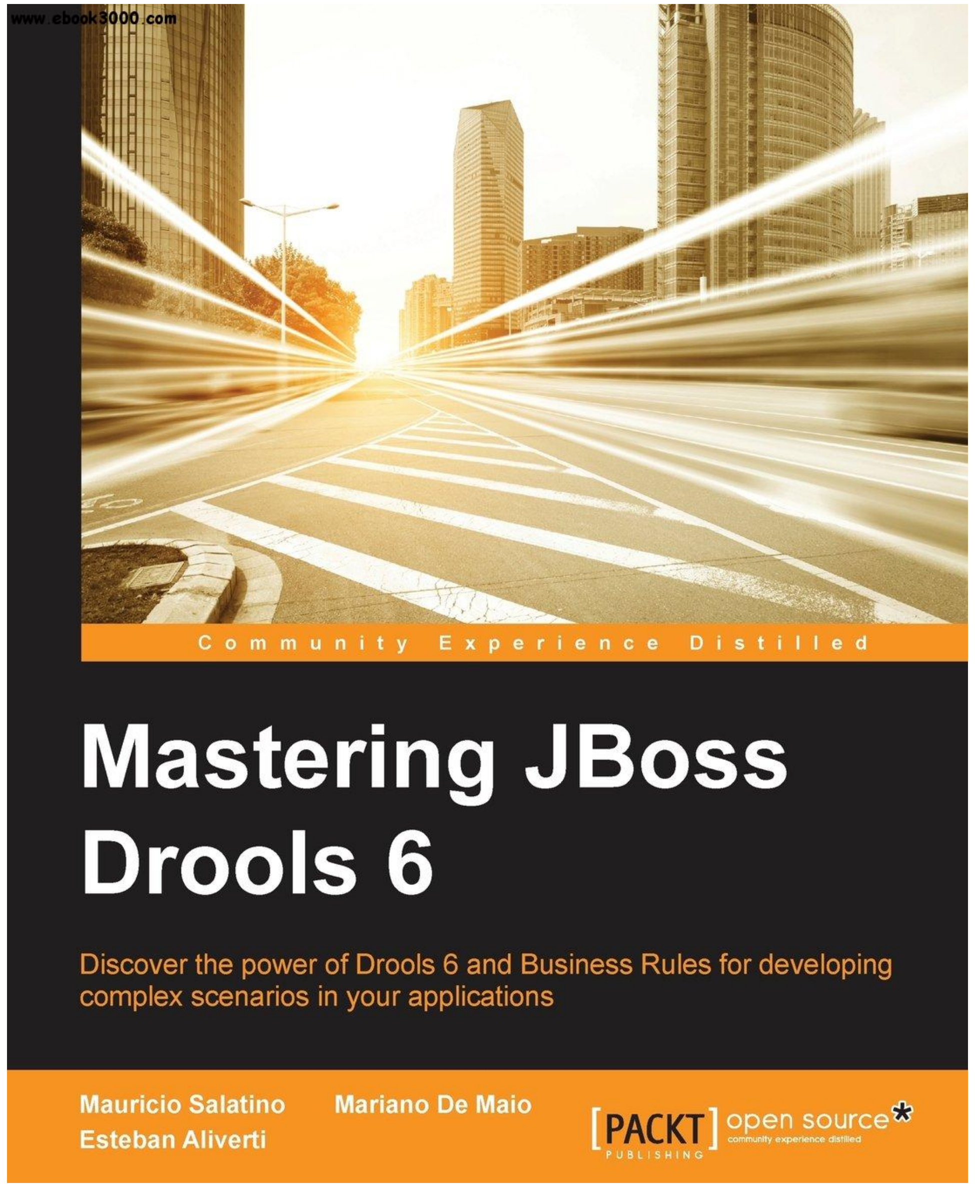
numberOfRules	useCanonicalModel	Units
100	false	9.291 ± 0.169 ms/op
100	true	0.209 ± 0.004 ms/op
500	false	44.774 ± 0.742 ms/op
500	true	0.926 ± 0.012 ms/op

Executable Model

Example (PatternTest.java)

```
Result result = new Result();
Variable<Person> person = DSL.declarationOf(Person.class);

Rule rule = rule("adult")
        .build( pattern( person ).expr( "exprA", p -> p.getAge() > 18),
                on(person).execute(adult -> result.setValue( "Person is adult: " + adult.getName()))
        );

KieBase kieBase = KieBaseBuilder.createKieBaseFromModel( new ModelImpl().addRule( rule ) );

KieSession ksession = kieBase.newKieSession();

ksession.insert(new Person("Leonardo", 3));
ksession.insert(new Person("Luca", 36));

ksession.fireAllRules();
assertEquals("Person is adult: Luca", result.getValue());
```

Books



Community Experience Distilled

Mastering JBoss Drools 6

Discover the power of Drools 6 and Business Rules for developing complex scenarios in your applications

Mauricio Salatino
Esteban Aliverti
Mariano De Maio

[PACKT] PUBLISHING open source* community experience distilled

The end

Questions?