Practical Rule Engines
Or
How I Learned to Stop Worrying and Set Up Us the Bomb

Thomas Meeks

This might not make sense without the corresponding talk.

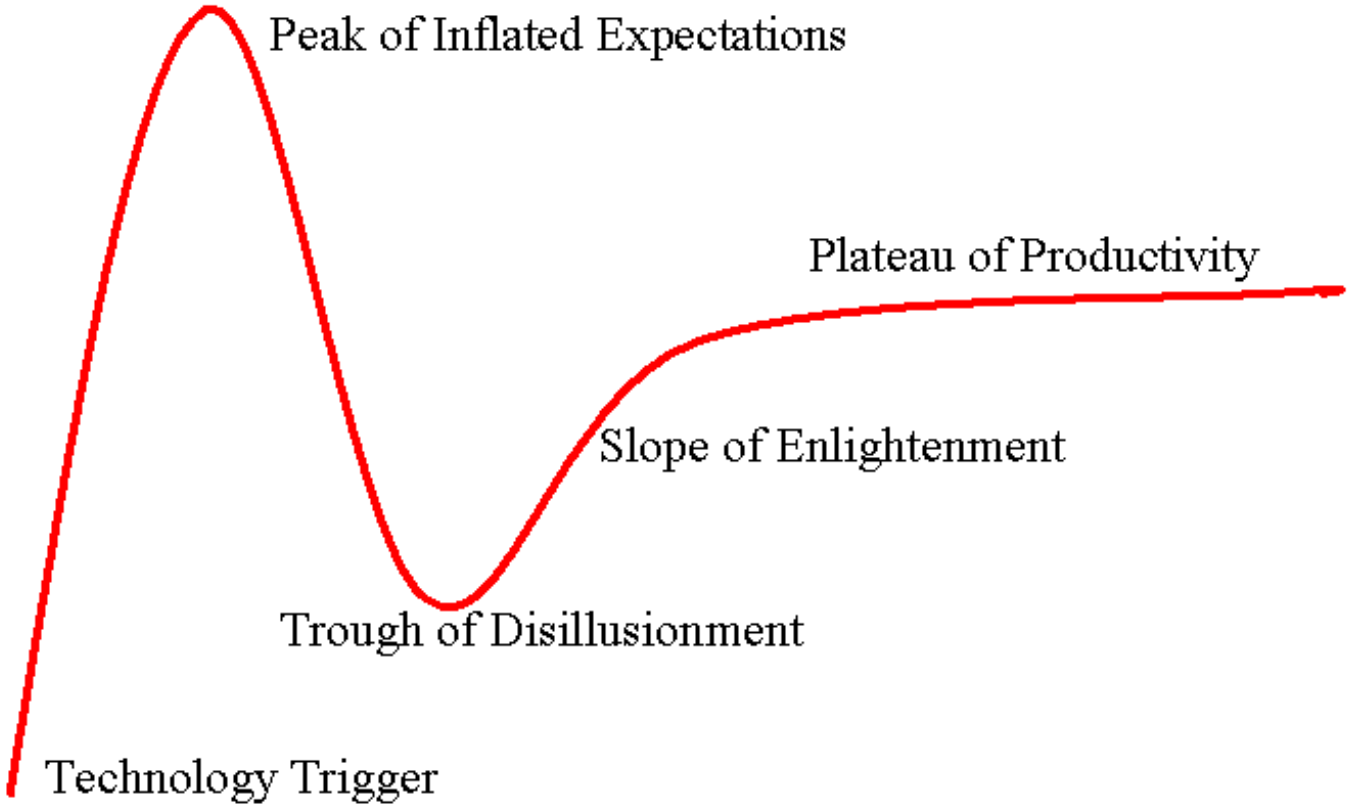
ORLANDO barcamp

What is a rule engine?

A system that attempts to act as a domain expert

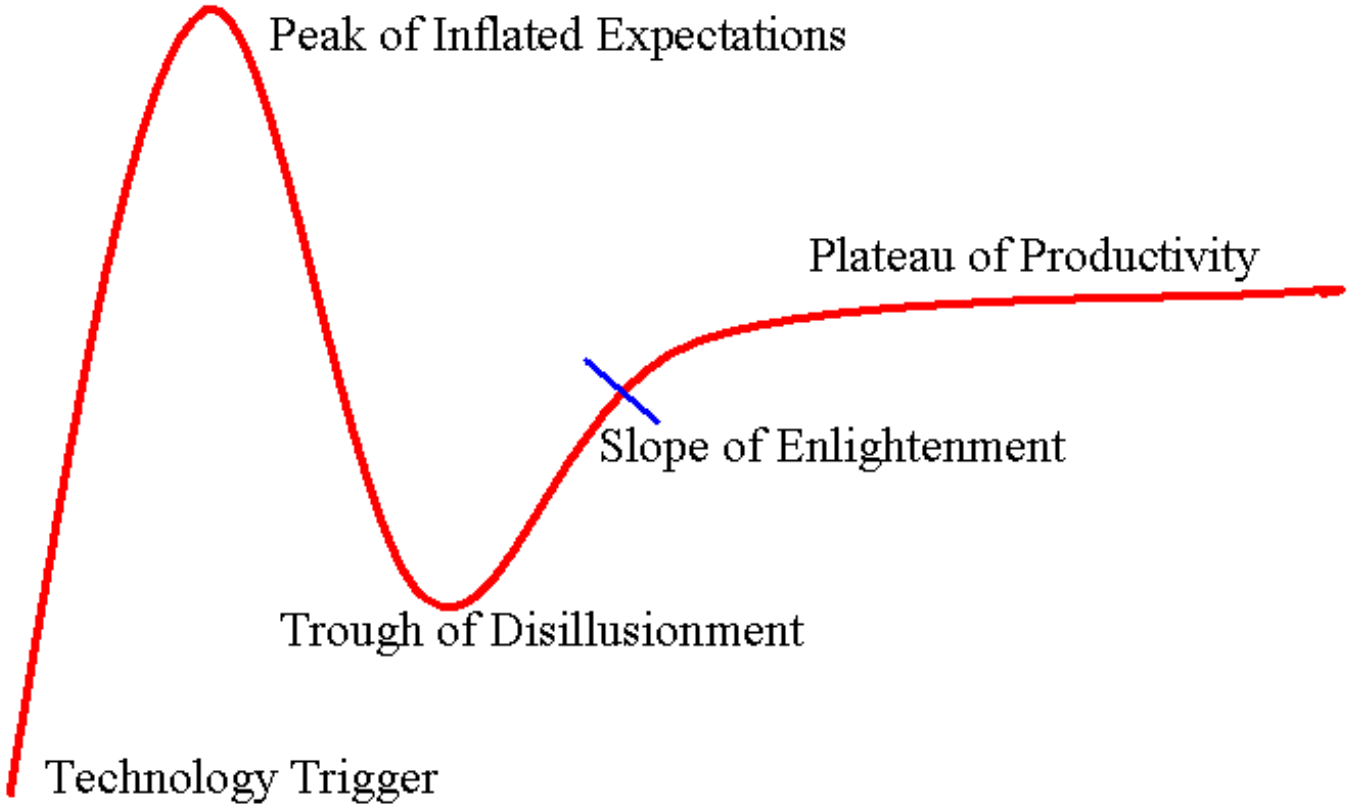
barcamp ORLANDO

Gartner Hype Cycle



barcamp ORLANDO

Gartner Hype Cycle



barcamp ORLANDO

So lets climb the Peak of Inflated Expectations

- Helps simplify complicated logic
- Lowers the cost of changing business logic
- Usually very fast
- Basically a business logic framework

barcamp ORLANDO

Wait, did you say... *framework?*

Form y *business logic?*

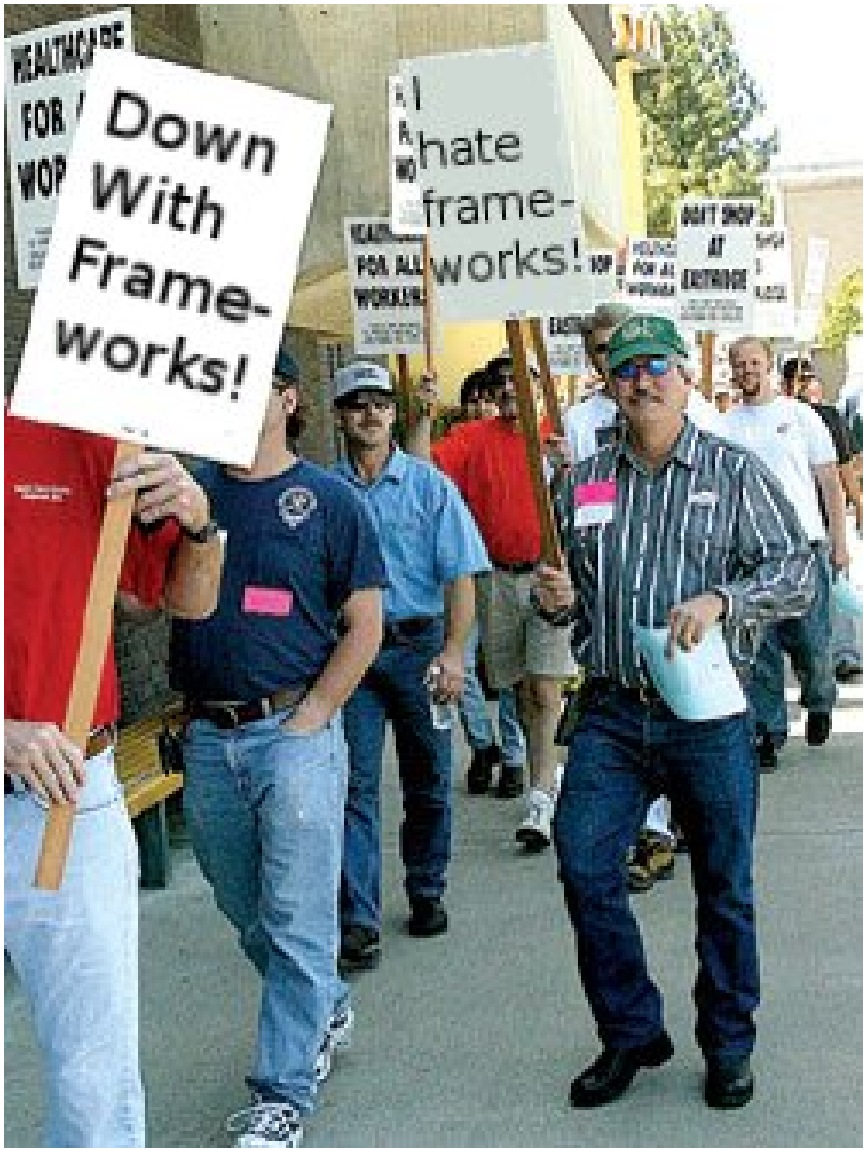
So I have a framework for:

- My UI (wicket, JSF, tapestry)
- My Database (hibernate, iBatis)
- Logging (commons logging)
- Compilation (ant, maven)

I even have a *framework* to tie together all my *frameworks*!!!! (spring, guice)

barcamp ORLANDO

And now you want me to use *another* framework... For my *business logic*!?



ORLANDO barcamp

I feel your pain (really), but...

It's a hammer, not a bullet

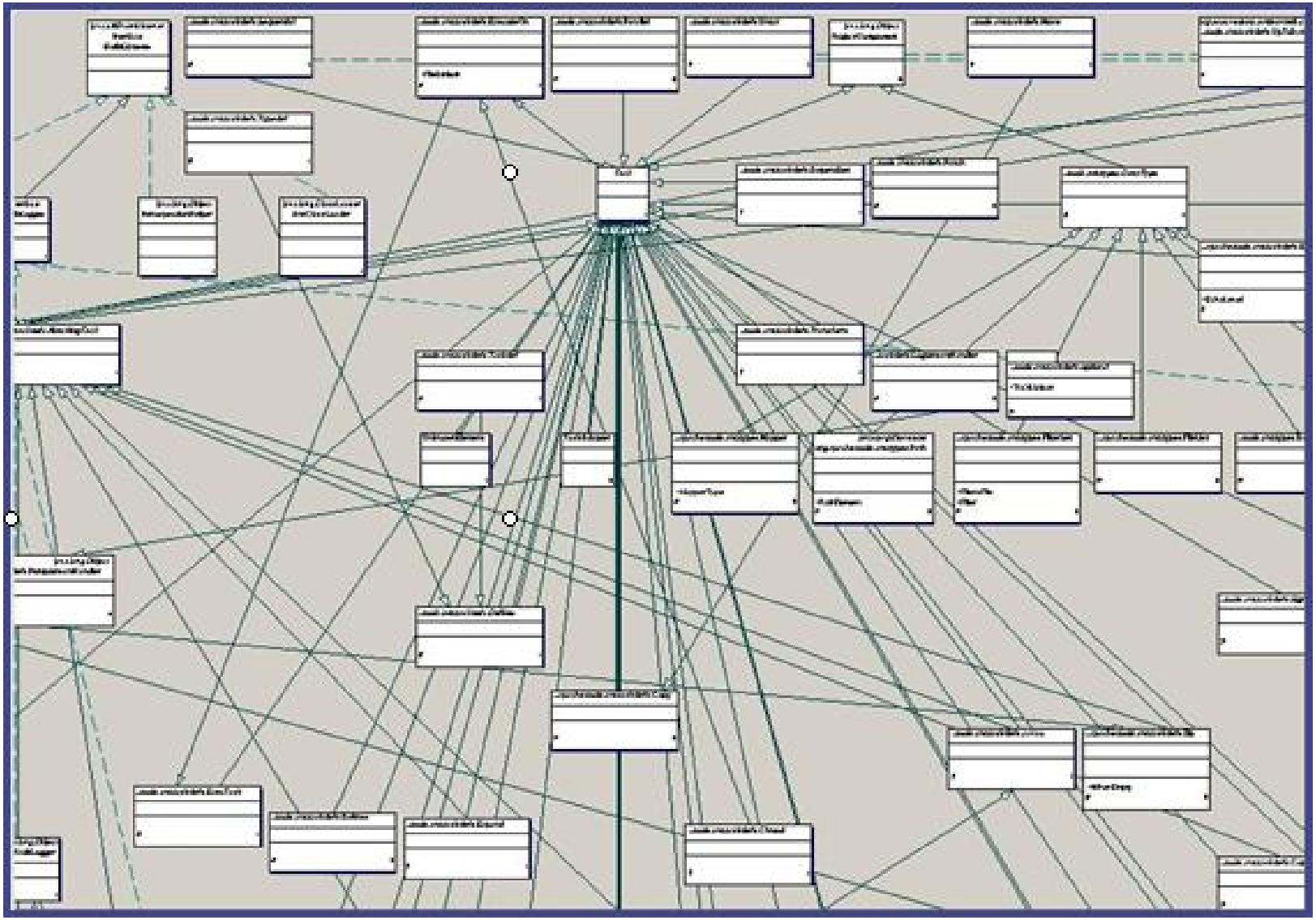
Use rule engines when it makes sense, or you will hate them

barcamp ORLANDO

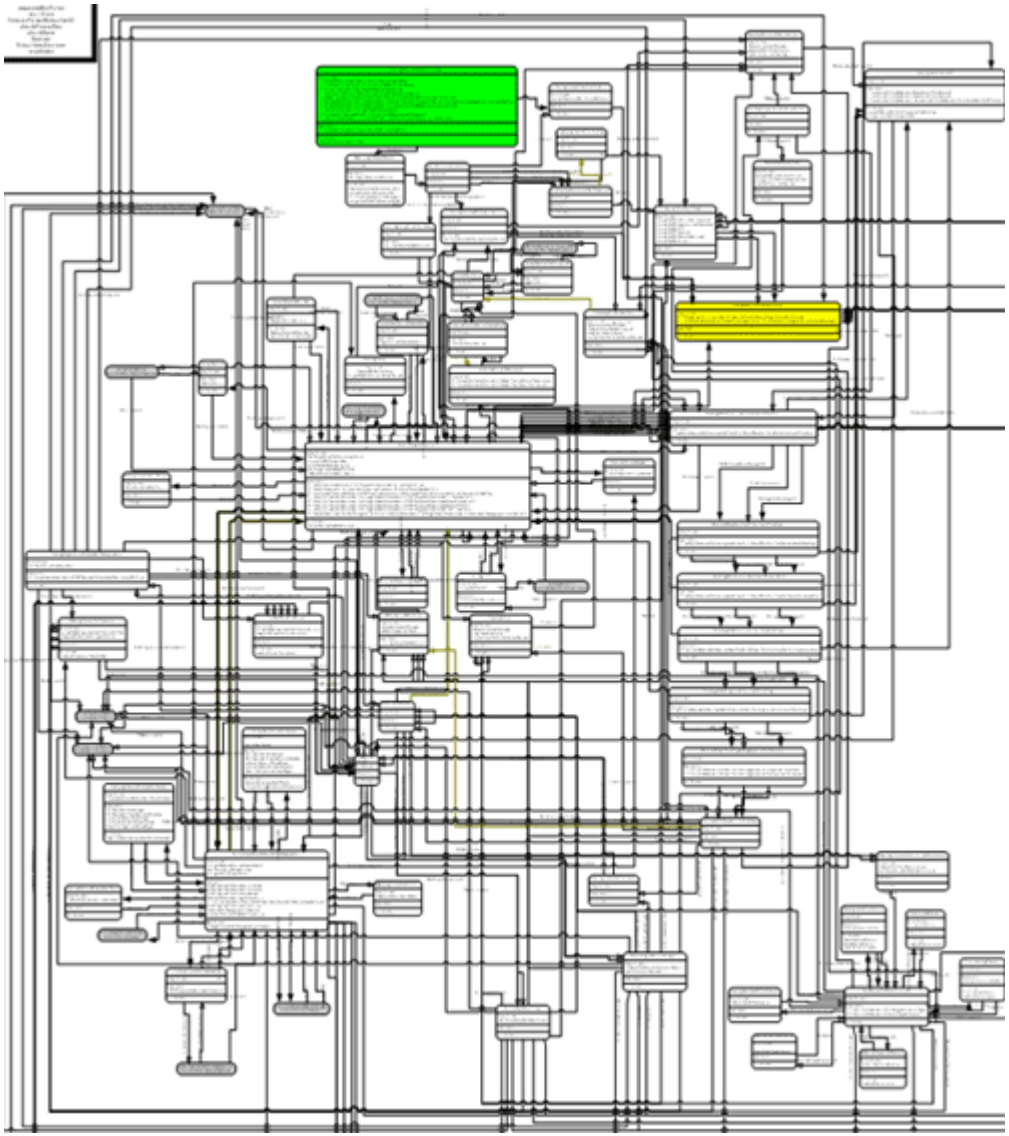
When should I use a Rule Engine?

- **Complicated** logic (*not* 1+1 = 2)
- **Changes** often (whatever *that* means)
- Traditional approaches are **unmaintainable**

barcamp ORLANDO



ORLANDO barcamp

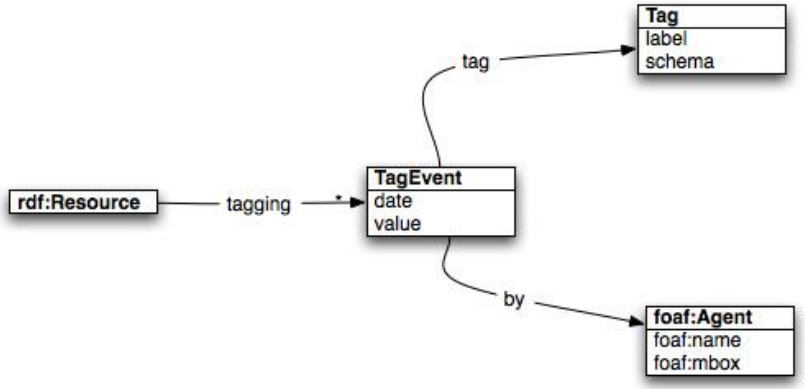


ORLANDO barcamp

Don't use for a shopping cart

- Don't use on applications with simple business logic
- If the logic is complicated but will never (**ever**) change, might not need it.
- Easy logic, changes often?
  - **Scripting**

barcamp ORLANDO



ORLANDO barcamp

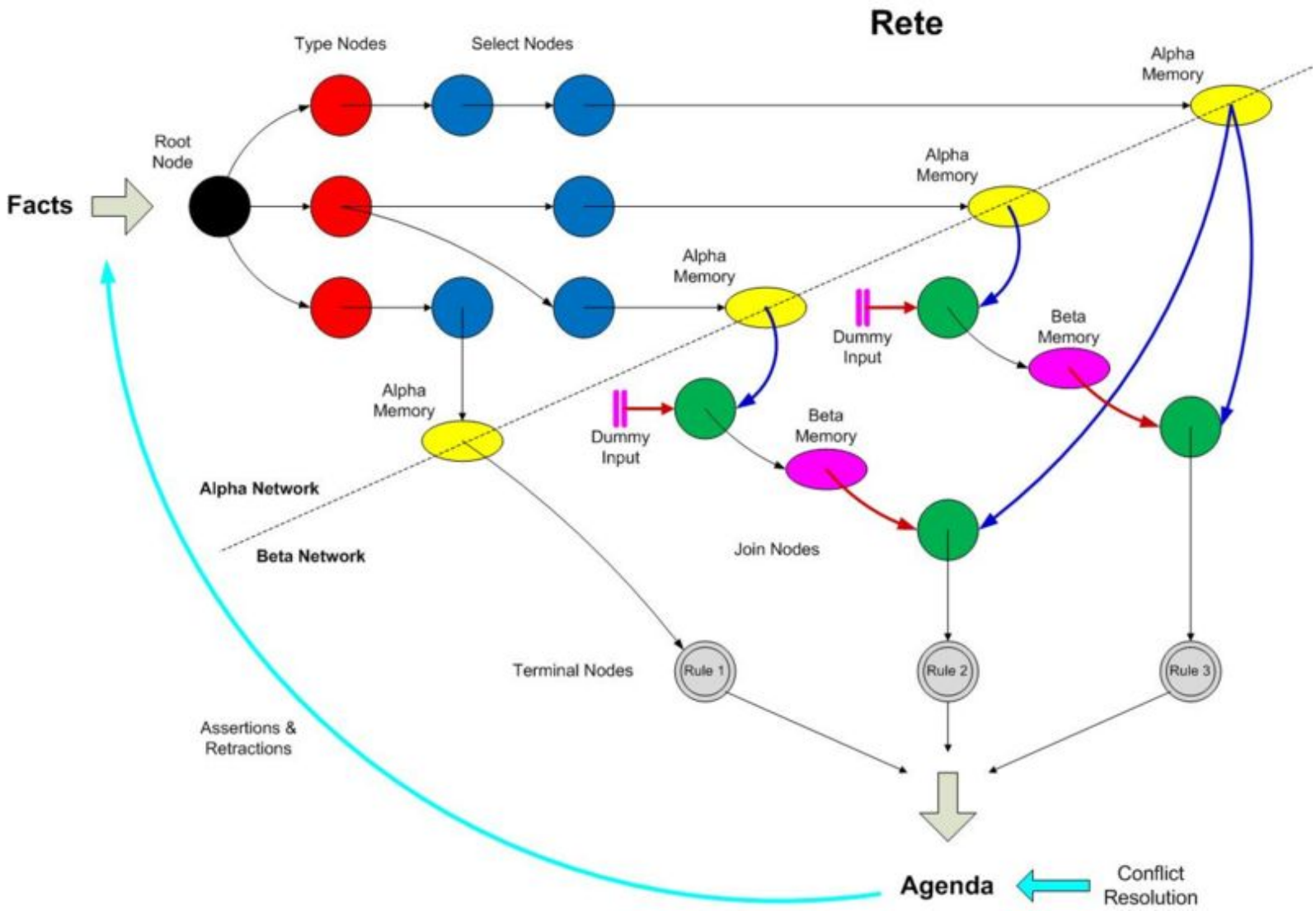
Didn't you say something about speed?

ORLANDO barcamp

Rete

- Latin for “net”
- As in network, or graph
- Performance is *theoretically* independent of the number of rules in the system
- Here's a quick overview

barcamp ORLANDO



ORLANDO barcamp

Example

ORLANDO barcamp

ORLANDO barcamp

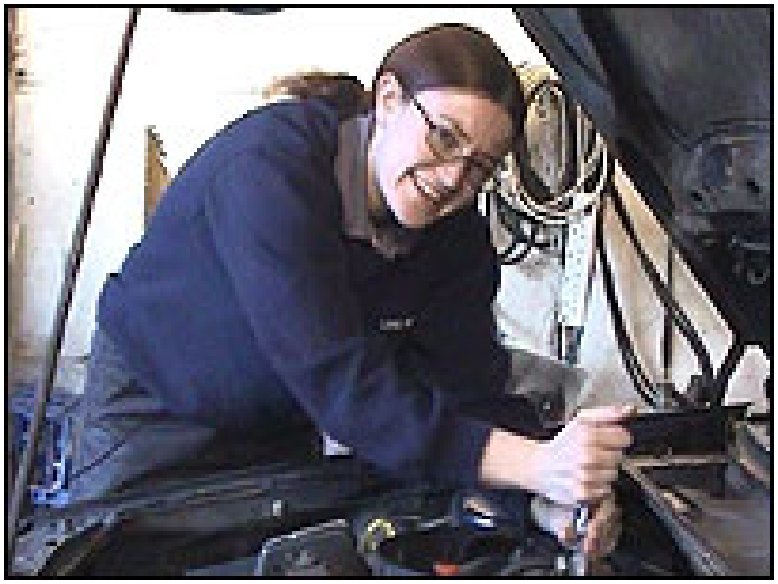
Captn' the Domain Expert

- Arr, Take off every ZIG!!! (For great justice! Grr.)

```
if (us.somebodySetUsUpTheBomb()) {
  for (Zig zig : us.getZigs()) {
    zig.launch(); // for justice!
  }
}
```

barcamp ORLANDO

Sally the Mechanic



- Zigs should not launch without their weapon system...

Which, of course, are...

Atomic Kittens!

ORLANDO barcamp

```
if (us.somebodySetUsUpTheBomb()) {
  for (Zig zig : us.getZigs()) {
    if (zig.hasAtomicKitty()) {
      zig.launch();
    }
  }
}
```

barcamp ORLANDO

Johnny the Regulator

- Zigs must have been inspected in the last 24 hours to launch!

ORLANDO barcamp

```
if (us.somebodySetUsUpTheBomb()) {
  for (Zig zig : us.getZigs()) {
    if (zig.hasAtomicKitty()
        && zig.isInspected()) {
      zig.launch();
    }
  }
}
```

barcamp ORLANDO

Johnny the Regulator

- Oh, and rule 23.43-57a#32.57 explicitly states that no more than 10 Zigs can fly at a time!

ORLANDO barcamp

```
if (us.somebodySetUsUpTheBomb()) {
  int i = 0;
  for (Zig zig : us.getZigs()) {
    if (i == 10) {
      break;
    }
    if (zig.hasAtomicKitty()
        && zig.inspected()) {
      zig.launch();
      i++;
    }
  }
}
```

barcamp ORLANDO

Captn' the Domain Expert

- Arr! Only 10!?
- If a Zig be shot down, launch more!
- Or ye be walkin' the plank!

ORLANDO barcamp

Hmmm....

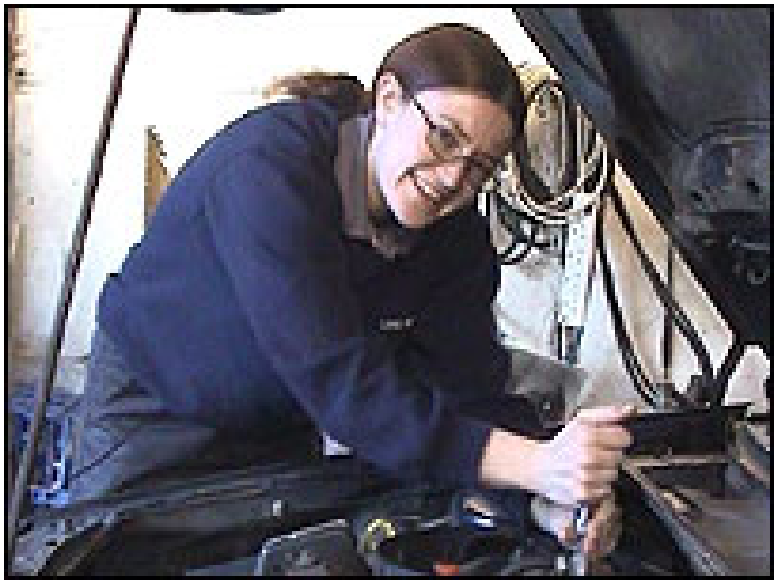
- We could keep checking every Zig's status in an infinite loop.
- We could implement the observer pattern on Zigs (when they explode, they tell someone).
- etc...
- Lets stick with the loop for the example

barcamp ORLANDO

```
int i = 0;
while (us.somebodySetUsUpTheBomb()) {
  for (Zig zig : us.getZigs()) {
    if (zig.hasExploded()) {
      us.getZigs().remove(zig);
      i--;
      continue;
    }
    if (zig.hasAtomicKitty() && zig.inspected()
        && i < 10) {
      zig.launch();
      i++;
    }
  }
}
```

barcamp ORLANDO

Sally the Mechanic



- If those Zigs get beat up, they should land so I can fix them!
- And don't try to launch them while I'm working on them!

ORLANDO barcamp

```
int i = 0;
while (somebodySetUsUpTheBomb()) {
  for (Zig zig : us.getZigs()) {
    if (zig.needsMaintenance()) {
      zig.land();
      mechanic.startFixing(zig);
      i--;
      continue;
    }
    if (zig.hasExploded()) {
      us.getZigs().remove(zig);
      i--;
      continue;
    }
    if (zig.hasAtomicKitty() && zig.inspected()
        && i < 10 && !zig.inMaintenance()) {
      zig.launch();
      i++;
    }
  }
}
```

ORLANDO barcamp

Johnny the Regulator

- I forgot to mention that rule 23.43-57a#32.57a explicitly states that all Zigs can fly if you fax form 453.438-347#B in triplicate

ORLANDO barcamp

Captn' the Domain Expert

- Arr! That form takes hours to fill!
- Arr! Launch 10 until we fax it, then Take off every Zig! (for great justice, grr.)

ORLANDO barcamp

```
int i = 0;
while (somebodySetUsUpTheBomb()) {
  form.asyncFillOutAndFax();

  for (Zig zig : us.getZigs()) {
    if (zig.needsMaintenance()) {
      zig.land();
      mechanic.startFixing(zig);
      i--;
      continue;
    }
    if (zig.hasExploded()) {
      us.getZigs().remove(zig);
      i--;
      continue;
    }
    if (zig.hasAtomicKitty() && zig.inspected()
        && (i < 10 || form.isFaxed()) && !zig.inMaintenance()) {
      zig.launch();
      i++;
    }
  }
}
```

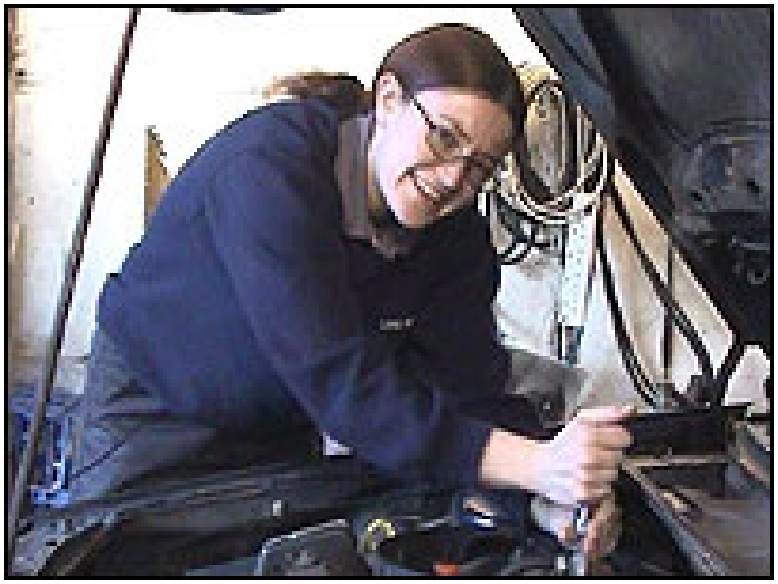
ORLANDO barcamp

Johnny the Regulator

- We just changed the rules!
- All Zigs must be pink to fly

barcamp ORLANDO

Sally the Mechanic



- Paint them pink!? That will take months! We have thousands!

ORLANDO barcamp

Captn' the Domain Expert

- Arr! Take off all pink Zigs!
- If we finish painting a Zig, launch it!

This is getting complicated!

- Thousands of Zigs? That loop could take a while.
- A lot of event-driven logic
- Looks like it is going to end up really messy
- No, the regulator will not stop

barcamp ORLANDO

Wait a second...

- You might be thinking – that was awfully arbitrary!
- Or – Hey, I can make up stupid requirements that are hard to implement in a huge loop cleanly too!
- I must assume you aren't in a regulated industry...

barcamp ORLANDO

So Lets Take a Look at Rules

- Specifically Drools / Jboss Rules
- It is a Domain Specific Language (DSL) that wraps plain Java
- Plenty of other implementations in other languages.
- Not necessarily more concise

barcamp ORLANDO

```
rule "take off every zig for great justice"
  no-loop true;
  when
    Us(somebodySetUpUsTheBomb == true)
    zig : Zig(inspected == true,
              pink == true,
              atomicKitten == true,
              inMaintenance == false,
              launched == false)
    launched : launched(launched < 10
                        || formFaxed == true)
  then
    zig.launch();
    launched.increment();
    update(zig);
    update(launched);
end
```

barcamp ORLANDO

```
rule "zig explodes"
  no-loop true
  when
    zig : Zig(destroyed == true)
    launched : Launched()
  then
    retract(zig)
    launched.decrement();
    update(launched);
end
```

barcamp ORLANDO

```
rule "repair zig"
  no-loop true
  when
    zig : Zig(needsMaintenance == true)
    mechanic : Mechanic()
    launched : Launched()
  then
    zig.land();
    launched.decrement();
    mechanic.startFixing(zig); //update zig when done
    update(zig);
end
```

barcamp ORLANDO

Where's the loop?

- It is implied
- Each rule fires for each fact combination that matches
- If you assert 1000 Zigs, the first rule will be checked 1000 times.

This is kinda confusing

- Think about it like an event-driven system
- It will re-schedule rules when insert(), update(), or retract() is called

barcamp ORLANDO

Neato Spandito

- New rules can be gracefully inserted into a large ruleset
  - Will automatically fire when conditions are met
- Caching facts makes execution *very* fast
- Properly managed rules can create very dynamic systems

barcamp ORLANDO

Now Lets Descend into Disillusionment

- Easy to overuse rules
- Not all logic (even in complex systems) should be in rules
  - Notice that I don't have a rule describing how a Zig launches
- Bugs can be evil

barcamp ORLANDO

Evil Bugs?

- Usually in a DSL – IDE's are not as mature
- Rules are recursive & loosely coupled by nature
- Rules usually compiled at runtime
  - But not every time they are executed

ORLANDO barcamp

Java Bug

ORLANDO barcamp

C / C++ Bug

barcamp ORLANDO

Drools Bug

barcamp ORLANDO

Onwards! To Enlightenment!

- Don't avoid adding fact classes
  - Trying to avoid it leads to messy rules
- Let the rule engine deal with the when, rather than the how
  - Orchestrate business logic
  - Work with your domain model as if it were a DSL itself

ORLANDO barcamp

Onwards! To Enlightenment!

- Assert object's, don't walk graphs
  - trade.getAccount().getClient() is usually bad
  - Helps as you add more rules
- Read!
  - A bunch of links are on www.thomasmeeks.com

ORLANDO barcamp

A quick aside on choosing a rule engine...

- Avoid XML DSL's
- It needs to have a NOT conditional (e.g. Operate on the non-existance of a fact)
- Don't trust claims of graphical programming
  - It is neat, but not a bullet
- Non-programmers probably shouldn't write the final rules in the rule engine

ORLANDO barcamp

Questions?

barcamp ORLANDO

Thanks!