RSpec Design Principles

Larry Peterson
Princeton University

Plea

The problem is already hard enough; resist the temptation to make it even harder.

or

Solve no problem before its time.

#1 “Thousand Flowers” Rule

Limit to minimal statement needed to reserve resources

- many “languages” to express researcher needs
- many “languages” to express device properties
- invent the best “matching process” and rule the world

#2 DIY Rule

Limit scope to operations that require privilege

- do allocate resources
- do not program/configure resources

#3 KISS Rule

Resist speculative feature creep

- reference implementation a prerequisite
- avoid excess precision
- simple specifications should remain simple

#4 Inheritance Rules

Leverage commonality and natural hierarchy

- support nesting/stacking of specifications
- consolidate commonality in base class(es)
- allow device specialization
- defining RSpecs is an on-going process

PlanetLab Strategy

- Classic PlanetLab gets by with a degenerate RSpec
  - all nodes equivalent / unspecified link (Internet)
- Ask the aggregate for right answer

```
DefaultRSpec = GetResources(Any)
Slice = CreateSlice(DefaultRSpec)
AllocatedRSpec = GetResources(Slice)
```

- XML is only a representation format
  - Define data model with Eclipse Modeling Framework (EMF)

Hide the Complexity

- In the data structure (the RSpec approach)
  - Tools help make simple things easy
- In the interface (the WSDL approach)
  - Ask the aggregate for its capabilities
  - Additional queries as needed