FlowOps: Open Access Network Management and Operation

MS Thesis Defense May 10, 2013 Matt Strum mstrum@flux.utah.edu

NETWORK MANAGEMENT IS HARD



OPEN ACCESS NETWORKS



AUTOMATION AND INTEGRATION



Current Solutions

- Still...
 - Vendor oriented
 - Task focused
- Examples
 - Junos Space Network Management
 - HP Automated Network Management
 - Cisco Prime

Method

AUTOMATION THROUGH RULES



Thesis Statement

An automated network management and operations framework built on a production rule system can capture the dependencies and relationships of both the network infrastructure and the role players in open access network environments.

Contributions

- Layered model defining entities and behaviors
 - Knowledge Store holding the entities
 - Rules Engine to enforce behavior
- Driver Engine for network device support
- **Prototype** demonstrating utility to all actors

Work Built On

- KnowOps [Chen et al. Hot-ICE11] Unification of:
 - PACMAN [Chen et al. ACM 5th 2009]
 - Network management workflow tasks
 - COOLAID [Chen et al. ACM 6th 2010]
 - Declarative language to capture knowledge from domain experts and documents
 - DÉCOR [Chen et al. SIGCOMM 2010]
 - Database-oriented network management system

1. Introduction

2. Architecture

- 3. Implementation
- 4. Evaluation
- 5. Related Work
- 6. Conclusion

System



Model





VIEWS



DRIVER ENGINE



Switch



WORKFLOW



ALLOCATION



ALLOCATION



ALLOCATION



ALERTS



ALERTS



- 1. Introduction
- 2. Architecture
- 3. Implementation
- 4. Evaluation
- 5. Related Work
- 6. Conclusion

Knowledge Store

Network Operator API Abstraction Layers Driver Engine ACME Driver ACME Acme Switch Infrastructure

- Service layer services
 - LAN, VLAN
 - E-LINE, E-LAN, E-TREE
- Network operator layer services
 VLAN
- API for allocation, views, etc.
- Uses Drools rule engine

Drools Rule Engine

Knowledge

Store / Rules Engine

Abstraction Layers



Driver Engine

Abstraction Layers Abstraction Layers Rules Engine ACME Driver OpenFlow Driver ACME ACME Switch Switch Infrastructure

Network Operat

- Commands
 - AddVlan, RemoveVlan
- Simulated driver
 - Acts as no-ops
 - Hosts API to simulate faults
- OpenFlow driver

- 1. Introduction
- 2. Architecture
- 3. Implementation

4. Evaluation

5. Related Work

6. Conclusion

Environment



Emulating the Network



Configuration



Configurations

- Used Mininet environment
- Services between various actors
- All combinations of service types



Example Service Definition

<resource_specification> <link type="ethernet"> <endpoint id="ServiceProvider:Port" /> <endpoint id="Bob:Port" /> </link> </resource_specification>

Fault Management

- Used simulated driver
- Brought port down



Fault Management

- Allocate path
- Bring non-critical port down
- Path automatically fixed
 - Notify actors about action



Fault Management

- Allocate path
- Bring critical port down
- Path unfixable
 - Notify actors about state



Evaluation

• FlowOps enables...

 Actor to infrastructure automation and integration in an open access network environment

- 1. Introduction
- 2. Architecture
- 3. Implementation
- 4. Evaluation
- 5. Related Work
- 6. Conclusion

Related Work

- PRESTO [Enck et al. IEEE 2009]
 - Configuration with transformation from high-level templates
- 4D [Greenberg et al. ACM SIGCOMM 2005]
 - Configuration through layers
- Network federation [Hayashi et al. AICT 2011]
 - Standardize services at the edges
- ChoiceNet [Rouskas et al.]
 - Expose several service layers

Related Work

- Generic Root Cause Analysis (G-RCA) [Yan et al. ACM 2010]
 - Analyze network events using rules
- Network-wide Information Correlation and Exploration (NICE) [Mahimkar et al. CoNEXT 2008]
 - Troubleshoot chronic network issues through analyzing statistical correlations

- 1. Introduction
- 2. Architecture
- 3. Implementation
- 4. Evaluation
- 5. Related Work

6. Conclusion

Contributions

- Layered model defining entities and behaviors
 - Knowledge Store holding the entities
 - Rules Engine to enforce behavior
- Driver Engine for network device support
- **Prototype** demonstrating utility to all actors

Conclusions

- Layered model simplifies logic
- Rule engine enables automation of...
 - Dependencies in our model
 - Network management tasks
 - Actor to infrastructure integration in an Open Access Network environment

Ongoing Work

- Enhancing prototype implementation
- Moving to test in lab environment

Future Work

- Major
 - More drivers, including traditional switches
 - Create sample services
- Expand network...
 - accounting, performance, and security
- Support more services at service and network operator layers

Acknowledgements

- EntryPoint LLC
 - Jeff Christensen
 - Robert Peterson

Thank you!

ANY QUESTIONS?

BACKUP SLIDES

Background

OPEN ACCESS NETWORKS

- Public
 - Utah's Utopia
 - Stockholm's Stokab
 - Amsterdam's CityNet
- Private
 - Reggefiber
 - Quadracom
 - MBC's network

Background

RETE ALGORITHM





DROOLS ENTRY-POINTS

Host

FactA

Entry-Point "N"

FactA

Knowledge Session

Default Store	"X" Store	rule "Rule 1"
Fact A	Fact A	when
		factA : FactA()
		<pre>factA : FactA() from entry-point "X"</pre>
Rules		

Default

Background

DROOLS CHANNELS



Background

DROOLS SALIENCE

Simultaneously triggered rules



ACTORS



SERVICES

SERVICES

