

ERGONOMIC ISSUES IN SYSTEM CONFIGURATION



Paul Anderson <dcspaul@ed.ac.uk>

[http://homepages.inf.ed.ac.uk/dcspaul/
publications/chimit-2008-talk.pdf](http://homepages.inf.ed.ac.uk/dcspaul/publications/chimit-2008-talk.pdf)

SYSTEM CONFIGURATION

SERVICE SPECIFICATION



WHAT IS INVOLVED ?

- How do we construct a system to provide the required service:
 - what software goes in each machine ?
 - what goes in all the “configuration files” ?
- Implement it:
 - load the software
 - create the configuration files
- Maintain it when things change:
 - either the requirements
 - or the system (failures)

WHY IS THIS HARD (1) ?

- It is large problem:
 - $O(1000)$ machines x $O(100-1000)$ parameters
- The relationships between all the internal services are complex:
 - DHCP, DNS, LDAP, Kerberos, IMAP, SMTP, AFS, NFS ...
- Things change all the time
 - failures are “normal”
- Mistakes are bad:
 - immediate service failures, or (perhaps worse) ...
 - hidden problems (security holes)
 - they are a very significant, real cost

WHY IS THIS HARD (2) ?

- Many different people are likely to be involved
 - they will have “overlapping concerns”
 - they will have different levels of skill
 - they may work for different organisations
- Deployment is a distributed process
 - it involves latency and uncertainty
 - asymptotic configuration
- Existing tools are primitive
 - designed for manual use at a “low-level”
 - no support for “top-down” configuration

CREATE A NEW WEB SERVER

- Configure the supporting infrastructure
 - create a DNS entry
 - create a DHCP entry
 - create holes in the firewall
 - create and sign SSL certificate
 - add to backup system
 - etc ...
- Configure the server
 - partition the disks, load the software etc ..
 - set up the services:
 - dns, networking, timeservice, apache, authorisation etc ..

LEVELS OF CONFIGURATION

1. “Copy this disk image onto these machines”
2. “Put these files on these machines”
3. “Put this line in sendmail.cf on this machine”
4. “Configure machine X as a mail server”
5. “Configure machine X as a mail server for this cluster” (and the clients to match)
6. “Configure any suitable machine as a mail server for this cluster” (and the clients to match)
7. Configure enough mail servers to guarantee an SMTP response time of X seconds

LEVELS OF AUTOMATION

- Fully automating the highest levels is not possible
 - generally, humans are responsible for the high levels, and the lower levels are automated
 - this forms one kind of interface (inter-level)
- Even at any one particular level, some aspects are handled by (different) people and automatic processes (intra-level)
 - multiple specialists deal with different aspects
 - different tools are used for different aspects
- A typical installation has a complex web of interactions
 - between different people and systems/tools

CONFLICTS

- One administrator (or automatic process) may specify a set of files to be copied onto some machines
- Someone (or something) else may decide on a different set
- How are people aware that there is a conflict?
 - do they have to communicate to resolve it?
- A uniform configuration tool helps
 - it will probably at least highlight the conflict
 - it might be able to resolve it
 - or at least, it might contain information enabling it to be resolved manually, without communication

CAN THE LANGUAGE HELP?

The language forces explicit values to be specified:

Aspect A

Use server Y

Aspect B

Use server X

This conflict is irreconcilable without human intervention because we don't know the intention

The user really only wants to say ...

Aspect A

Use any server on my Ethernet segment

Aspect B

Use one of the servers X, Y or Z

These constraints can be satisfied by using Y (assuming Y is on the right segment)

LOOSE SPECIFICATIONS

- Loose specifications can help when many people are involved in defining a configuration
 - these can also arise from automatic sources
- But there are some issues -
 - there is a danger of overspecification
 - there is a danger of underspecification
 - the system needs to operate at a fine-enough granularity
 - there can be some degree of unpredictability in the outcome
 - issues of responsibility and security are harder to resolve

REFINEMENT

- Typically, humans need to make the high-level decisions
 - these are manually translated into some lower-level specification (think of the web-server example)
- This kind of refinement can be done automatically
 - it requires trust in the system
 - the system needs to be able to explain the decisions
- At a higher level planning becomes more significant
 - For example, to change a server from A to B
 - deploy server B
 - change all clients to point to B instead of A
 - undeploy server A

A HYBRID APPROACH

- The appropriate point for the interface between manual and automatic procedures cannot be fixed
- Different sites have different priorities and resources available -
 - small or large numbers of people
 - a need for autonomic response
 - a cautious approach to changes
 - etc ...
- And these will vary across the different components

MEDIATED CONFIGURATION

- A suitable framework might allow ...
 - humans to make decisions
 - decisions to be deferred to other (remote) humans
 - intelligent modules to make automatic decisions
 - humans to vet automatic decisions
 - automatic rules to validate human decisions
 - intelligent modules to suggest appropriate options
- These processes may apply to deployment processes, as well as the desired configurations
- Tools such as this have been used to plan, for example, disaster response operations (I-X)
 - <http://www.aiai.ed.ac.uk/project/ix/>

I-X

- Process panels
 - similar to to-do lists that people use to plan and monitor processes
- A planning system
 - helps to explore different courses of action using what-if scenarios
- A domain modelling tool
 - to produce a library of best-practice or common ways of doing things
- A messaging system to help communication
- A tool for managing organisational relationships between panels

Issues

Description	Annotations	Priority	Action

Activities

Description	Annotations	Priority	Action
▼ move medical-box1 60 70		▼ Normal	▼ Refined by move-object
▼ select-transport-at m...		▼ Normal	▼ Refined by use-helicopter
seek-authority super...	done - succ...	▼ Normal	▼ Escalate to Supervisor
load Heli1 medical-box1		▼ Normal	▼ Refined by load-object-onto-vehicle
travel Heli1 60 70		▼ Normal	Refined by load-object-onto-vehicle
unload Heli1 medical-...		▼ Normal	Done

State

Pattern	
longitude medical-box1	47
latitude medical-box1	47
type medical-box1	thing
type Truck1	truck
longitude Truck1	47
latitude Truck1	47
type Heli1	helicopter
longitude Heli1	47

- Refined by load-object-onto-vehicle
- Done
- N/A
- Satisfy conditions
- Escalate to Supervisor

Annotations

Key	Value

ERGONOMIC ISSUES IN SYSTEM CONFIGURATION



Paul Anderson <dcspaul@ed.ac.uk>

[http://homepages.inf.ed.ac.uk/dcspaul/
publications/chimit-2008-talk.pdf](http://homepages.inf.ed.ac.uk/dcspaul/publications/chimit-2008-talk.pdf)