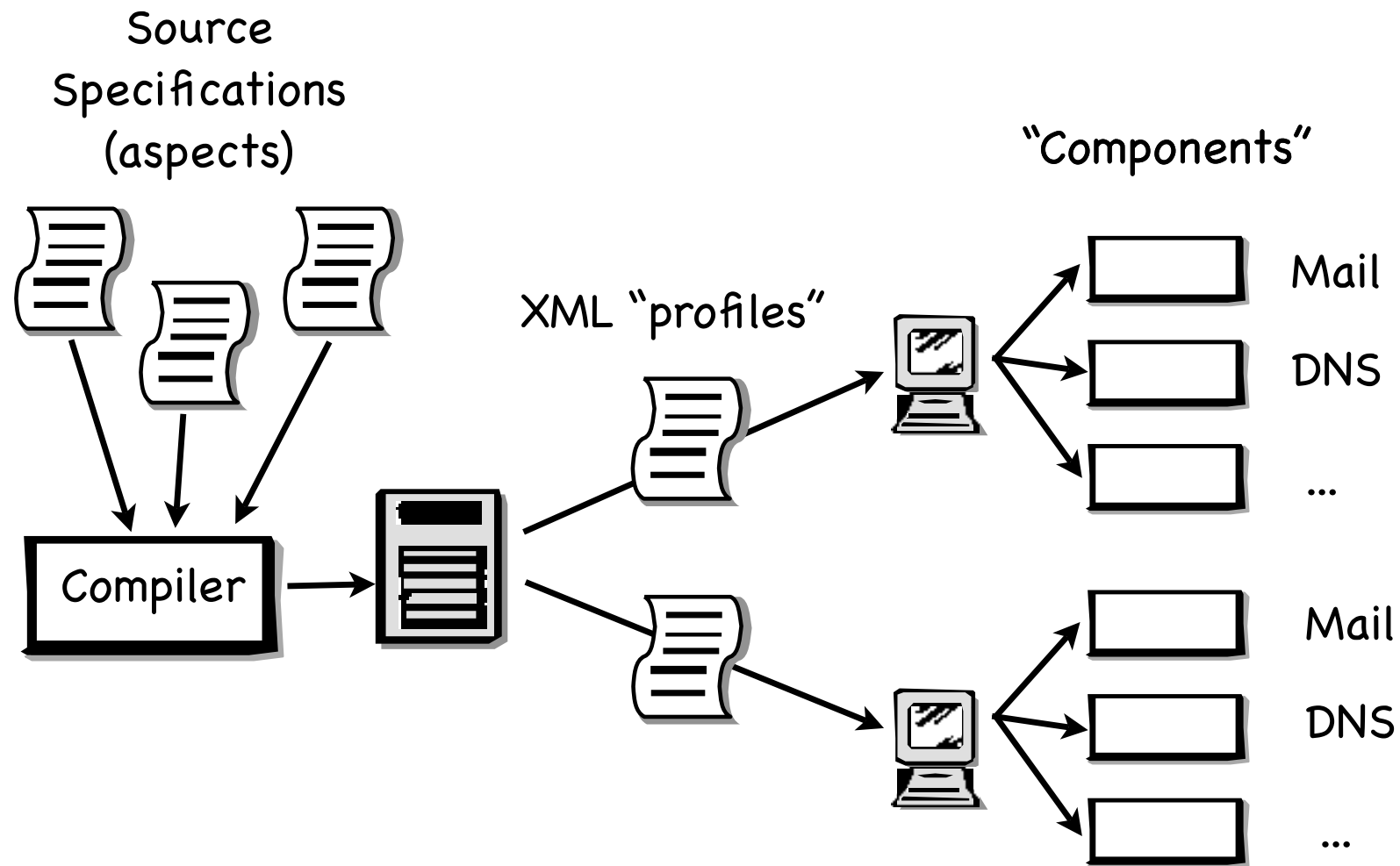


Configuration Validation With LCFG

Paul Anderson <dcspaul@inf.ed.ac.uk>

- A rapid overview of LCFG
- Current and potential validation in LCFG configurations
- Correctness by design

LCFG Architecture



- The profile usually contains all the configuration information necessary to distinguish a client from any other client (“proscriptive” configuration).
- The profile contains “resources” in a simple, uniform format. The components translate this into machine-specific configuration files.
- The sources are “aspects” - they contain information about (intersecting) classes and about relationships (as well as information about individual hosts).
- The compiler translates the aspects into individual profiles.

Example Profile

```
#include <dcspaul/nine3/sitedefs.h>
#include <dcspaul/nine3/options/dns-server.h>
#include <dcspaul/nine3/options/printing.h>
#include <dcspaul/nine3/options/vigor.h>
#include <dcspaul/nine3/options/webserver.h>
....
#include <lcfg/hw/dell_optiplex_gx270.h>
#include <lcfg/options/video_nvidia.h>
....
mailng.relay      mail.inf.ed.ac.uk
....
```

Validation in LCFG

- The LCFG compiler can validate individual resources at compile time.
- However, it is the relationships between resources which are usually more interesting/important.
- The LCFG source specifications contain all the necessary information to validate these (important!).
- At a real site, production configurations are usually “staged”, so compiled “pre-production” configurations will be available for validation.

Some Possible Validations

- All clients are running the LCFG mail component and define a mail server.
- All clients have the sendmail package installed.
- Anything which is referenced as a mail server (by any client) is classed as a mail server (running a server configuration).
- Anything which is classed as a mail server has a corresponding hole in the firewall for SMTP.
- Etc

Why Don't We Validate?

- I'm not sure. Maybe
- What do we decide to validate? Detailed validation rules could be many times the size of the configuration itself? How do we know what is worthwhile?
- The bulk of our LCFG configuration is concerned with host and service configuration, as opposed to network configuration - this probably entails less complex dependencies (?)
- I'm interested in configuration systems which encourage correct configurations "by design" ...

Correctness by Design

```
10 LET F=1
20 IF N>1 THEN GOTO 60
30 LET F=F*N
40 LET N=N-1
50 GOTO 20
60 STOP
```

```
for (f=1;n>1;--n) {
    f=f*n;
}
```

```
fun fac n =
  if n<1 then 1
  else n*fac(n-1);
```

- Host A runs a sendmail server configuration
- Host B runs ipchains
- Host B has an SMTP hole to host A
- Etc ...

- Host A is mail server
- All mailservers run a sendmail server config
- Host B is a firewall
- All firewalls run ipchains
- All firewalls have SMTP holes to all mail servers
- Etc

Configuration Validation With LCFG

Paul Anderson <dcspaul@inf.ed.ac.uk>

- A rapid overview of LCFG
- Current and potential validation in LCFG configurations
- Correctness by design