Centralized Configuration Management of Distributed System on Value-added Service Platform

Tuomas Lemminkäinen
Supervisor: Prof. Jorma Jormakka
HUT, Networking Laboratory
Instructor: M.Sc.(Tech) Heikki Hakkarainen

Contents

• Background
• The problem of configuration management
• The distributed environment
• Existing solutions
• Created solution
• Conclusion
Background

- The average size of telecom operator is rising
- The size of installed value-added service platforms are rising
- Need for enhanced management

The Problem of Configuration Management

- Configuration definition
  - Traditionally linked ini-files
  - Object structure suggested
- Configuration validation
  - Parameter datatype definition
  - Parameter value validation
  - Parameter dependency definition and validation
- Applying configuration changes
  - Configuration data distribution
  - Configuration change notification
The Distributed Environment

• Heterogenous environment
  – Distributed service creation on specialized network elements
  – End-user-services are created by co-operation of multiple internal services

• LAN connectivity
  – Fast connections between the network elements
  – Internal network isolated from the public networks

• Common middleware environment
  – Internal resource management and location registry

Existing Solutions

• SNMP
  – De-facto management standard of the Internet
  – Distributed concept
  – Concentrates on the management of a single host
  – Change notification is problematic

• Commercial solutions
  – Based on management protocols like SNMP or a proprietary management daemon
  – Expensive
The Configuration Management System

• Centralized management
  – Centralized interface for changing the system configuration
  – Centralized configuration data storage on configuration servers
  – A public CORBA interface to the configuration
  – A management facade for integrating external management

• 3-tier architecture
  – Separated client, server and data-server tiers
  – Makes possible to create multiple implementations of each tier
  – Well defined interfaces between the tiers

Configuration Definition

• Configured entities are presented as objects
  – Type defined for each configured entity
  – Datatype and constraint definitions for each parameter
  – Configuration validation against the type definition of the object

• Configuration trees
  – The instances of the configured entities are bound on the configuration trees
  – Structural inheritance with overwrite method used on the trees
  – Abstract entities can be created for common parameters
  – Service specific parameters are separated
Conclusion & Future Work

• Configuration validation reduces errors
  – The developer has best knowledge on possible parameter values
  – Adjusting development to enforced configuration definition takes time

• Object definition forces the creation of configuration definitions

• Feature enhancements
  – Configuration wizards

Q & A