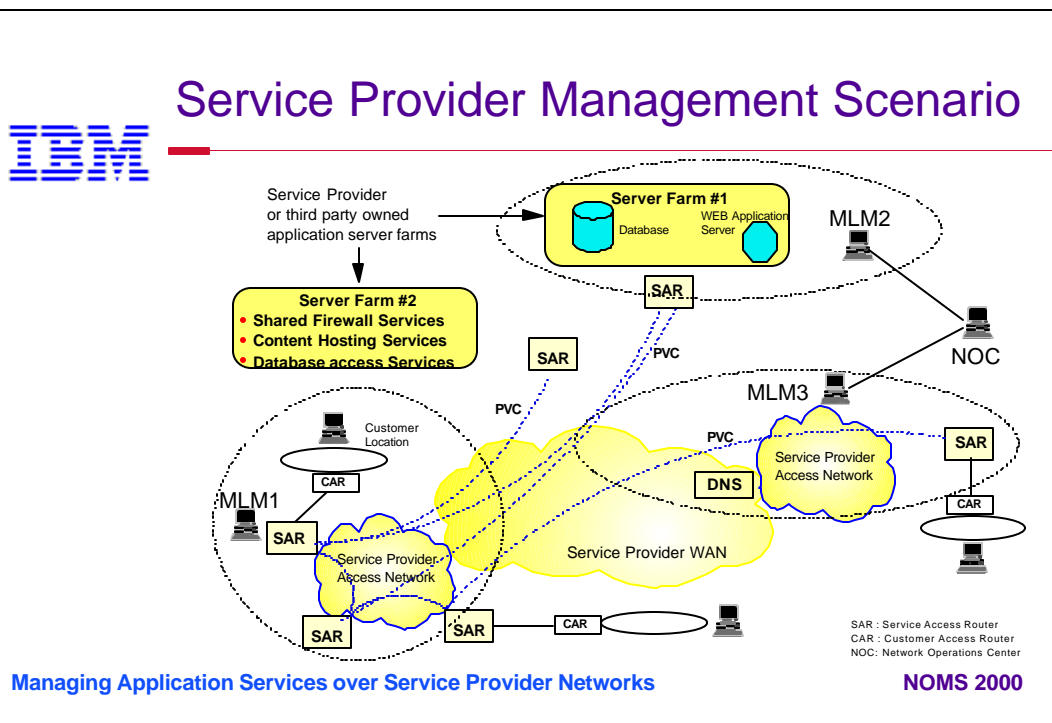




Managing Application Services over Service Provider Networks: Architecture and Dependency Analysis

Gautam Kar, Alexander Keller, Seraphin B. Calo
IBM T.J. Watson Research Center
April 11, 2000





SP Scenario: Issues and Requirements

□ Root-cause Analysis

- Determine a particular customers' service status,
- **Today:** manual test of services by operator, service dependencies not explicitly specified
- **Needed:** "drill-down", (automated) traversal of layers to find cause

□ Impact Analysis

- Determine which services/customers are affected by problem
- **Today:** Planned outages: Proactive warnings, Accidental outages: N/A
- **Needed:** "drill-up", determine potentially affected services/customers

□ Requirements:

- Make service interdependencies explicit and available
- Dependency model must allow upward and downward traversal
- Basis of vulnerability analysis for availability planning



Dependency Analysis: Design Issues

□ Application Management is not "Application Debugging"

- Focus on "externally visible" application parts (processes, SW package)
- Coarse-grained Problem Resolution (restart / reboot / reinstall)

□ Make use of already existing information:

- Assume no application-specific instrumentation, lightweight

□ Software components have to be registered in repository

□ Two kinds of dependencies:

- **Inter-system** (horizontal) dependencies:
 - Provide information on client/server relationships
 - needed for end-to-end problem determination
 - Example: "Resolver (DNS client) is bound to DNS server"
- **Intra-system** (vertical) dependencies:
 - Occur within a single system
 - Example: "WWW service requires Name service"

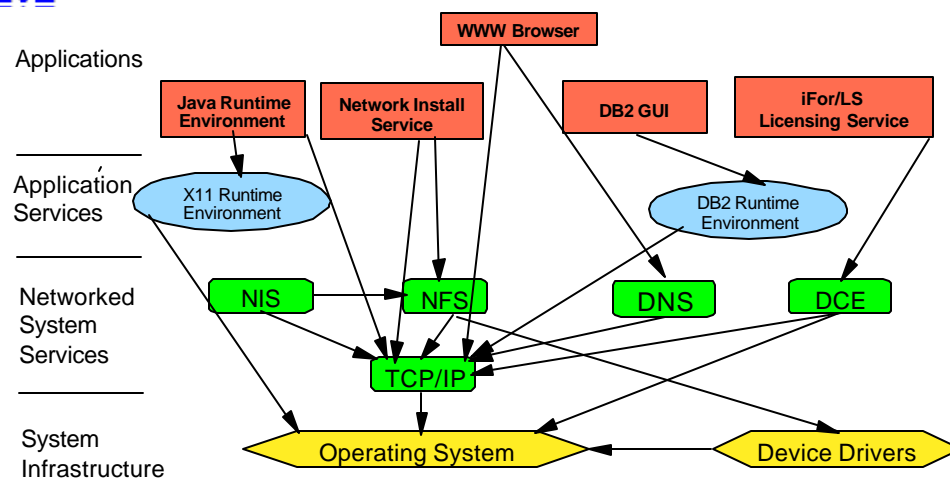


Dependency Analysis: The Approach

- **Pragmatic, Repository-based Approach:**
 - based on AIX ODM & Linux RPM (RedHat Package Manager)
 - yields information on dependencies between installed Software
- **A simple Example from AIX ODM:**
 - LPP & package names, install status, version #, fix#, description...
 - "Prerequisites" field of `bos.net.tcp.server`:
 - "bos.rte 4.2.0.0, bos.net.tcp.client 4.1.0.0"
- **This information tells us that:**
 - there is an *intra-system* dependency regarding the OS
 - "The TCP server requires (at least) AIX version 4.2.0.0"
 - there is an *inter-system* (client/server) dependency
 - "Every TCP client needs a working TCP server"

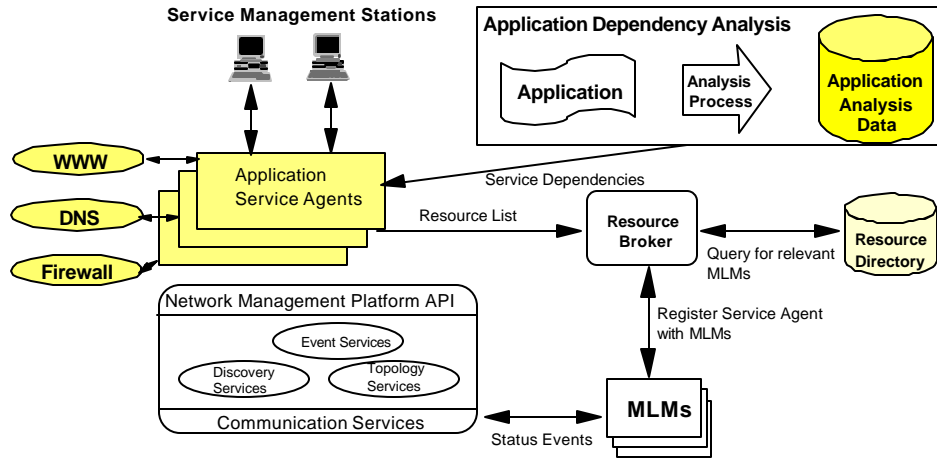


Generated Functional Dependency Model





Components of the Architecture

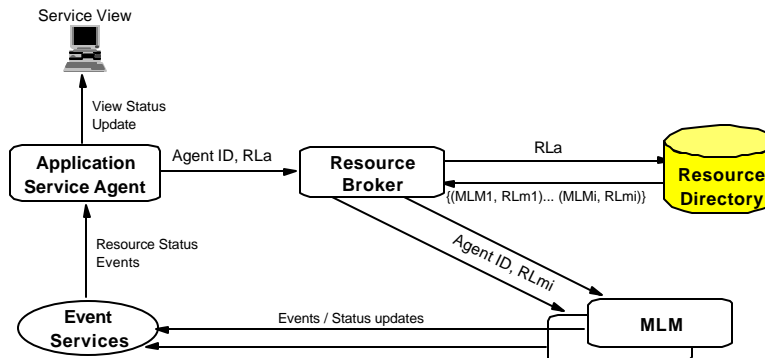


Managing Application Services over Service Provider Networks

NOMS 2000



Information Flow between Components



Managing Application Services over Service Provider Networks

NOMS 2000



Implementation Issues

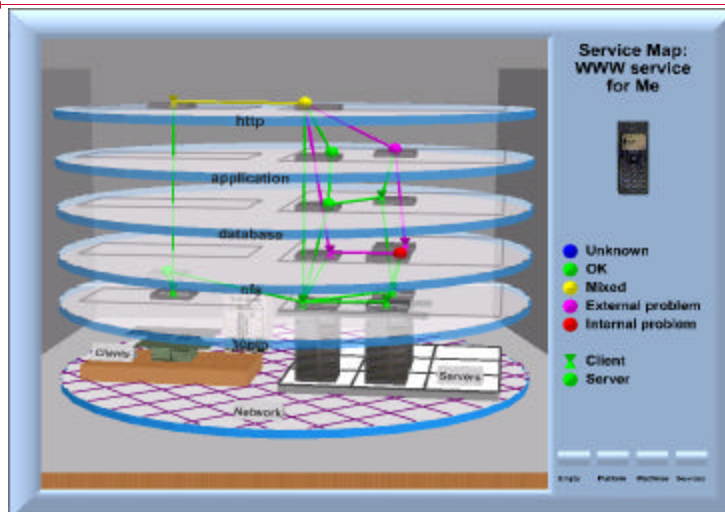
- **Dependency Model derived from Common Information Model**
 - Based on CIM Core, System and Application Schemas
 - Addresses packages, applications, services, subcomponents and their relationships (containment, dependencies, potential bindings)
- **Prototype components implemented in Java**
 - Communication through Java Remote Method Invocation
 - Existing Repository interfaces shielded by Java Native Interface
- **Enhanced Repository for Operating Systems (AIX, Linux)**
 - Single place for determining dependency information
 - Includes missing information
 - links processes to services
 - Looks up service / port mappings
 - Contains links to further configuration information

Managing Application Services over Service Provider Networks

NOMS 2000



Prototype GUI: Design Study



Managing Application Services over Service Provider Networks

NOMS 2000



Conclusions and Outlook

□ Results:

- Automated determination of static dependencies
- Obtained from operating system repositories
- Identifies functional dependencies between application services
- Small extensions to determine structural (potential) dependencies
- Distinction between instances hard, identification of effective bindings missing

□ Current work:

- Address runtime aspects of applications
- Model state transitions according to application lifecycle
- Extend CIM Application Schema
- Integration with available CIM Object Manager