Injecting Management

Robert Filman

Intelligent Systems Center
Advanced Technology Center
Lockheed Martin Missiles & Space
bob.filman@lmco.com

Microelectronics & Computer
Technology Corporation
filman@mcc.com

Management

- **Dynamic management of component-based systems**
  - **What one can do while a system is running to guarantee that policies are being followed**
- **Enabling policies in component-based systems**
  - **Without altering the component contracts**

Object Infrastructure Project

- **Goal:**
  - **Simplify the creation and evolution of systems composed of distributed components**
- **Observation:**
  - **Components perform localized, functional behavior**
  - **Separately generated components don't support system-wide properties: Ilities**
    - **scaleability, reliability, affordability, understandibility, quality of service, degradability durability, accountablity, flexiblity, security, manageability, ...**
- **Method:**
  - **Separate application coding from ilities**
  - **Automate going from ility specification (policies) to running composed systems**
    - **for certain ilities**

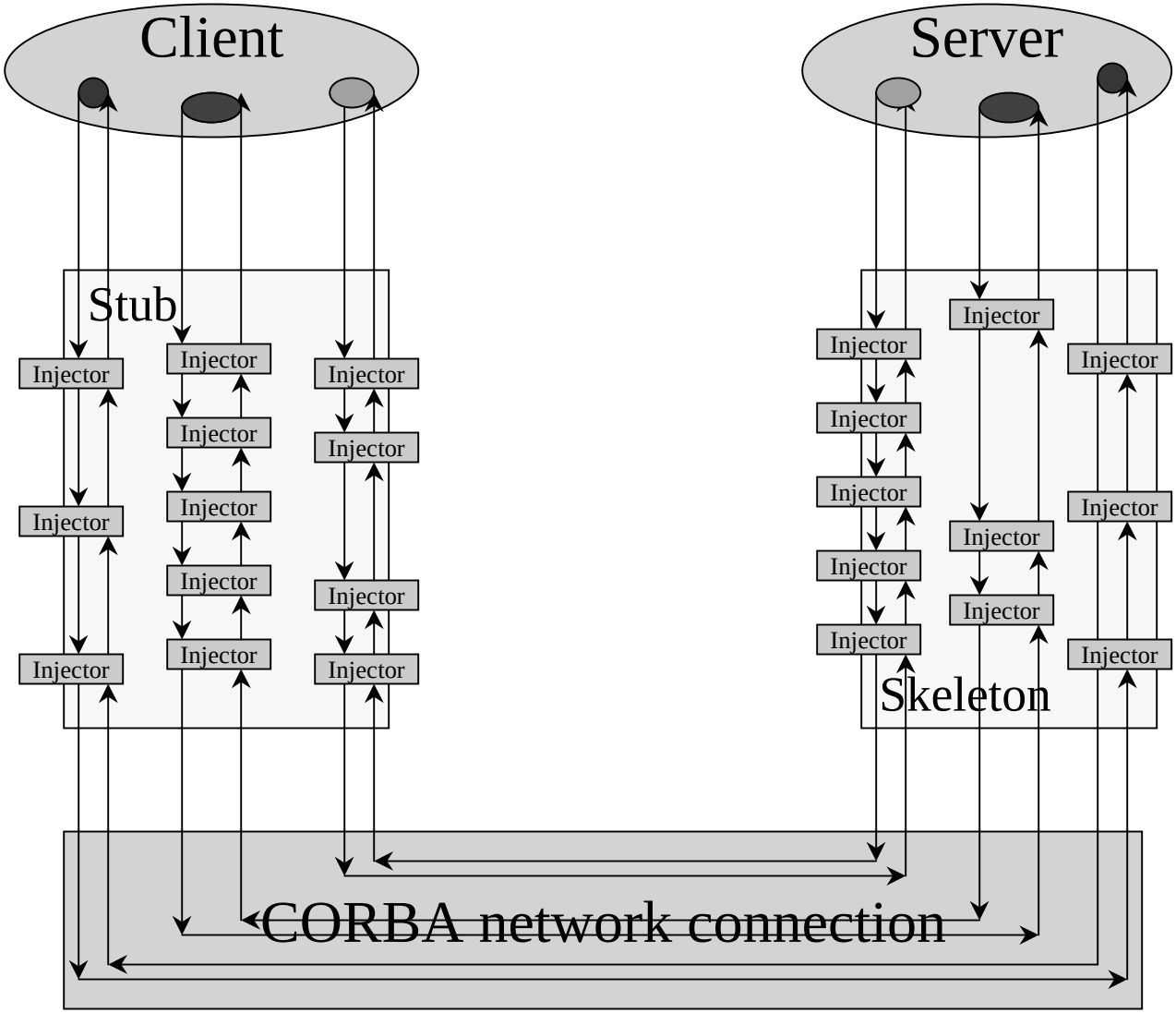
Insight

- **Ilities can be achieved by inserting behavior (services) into the communication paths between functional components.**

OIF (Object Infrastructure Framework)

- **Technology:**
  - **Wrappers around system components to implement ilities**
    - **Proxy/stubs**
    - **Injector-based**
  - **Automatic generation of wrappers from specifications**

Injecting Services



Key Framework Functions

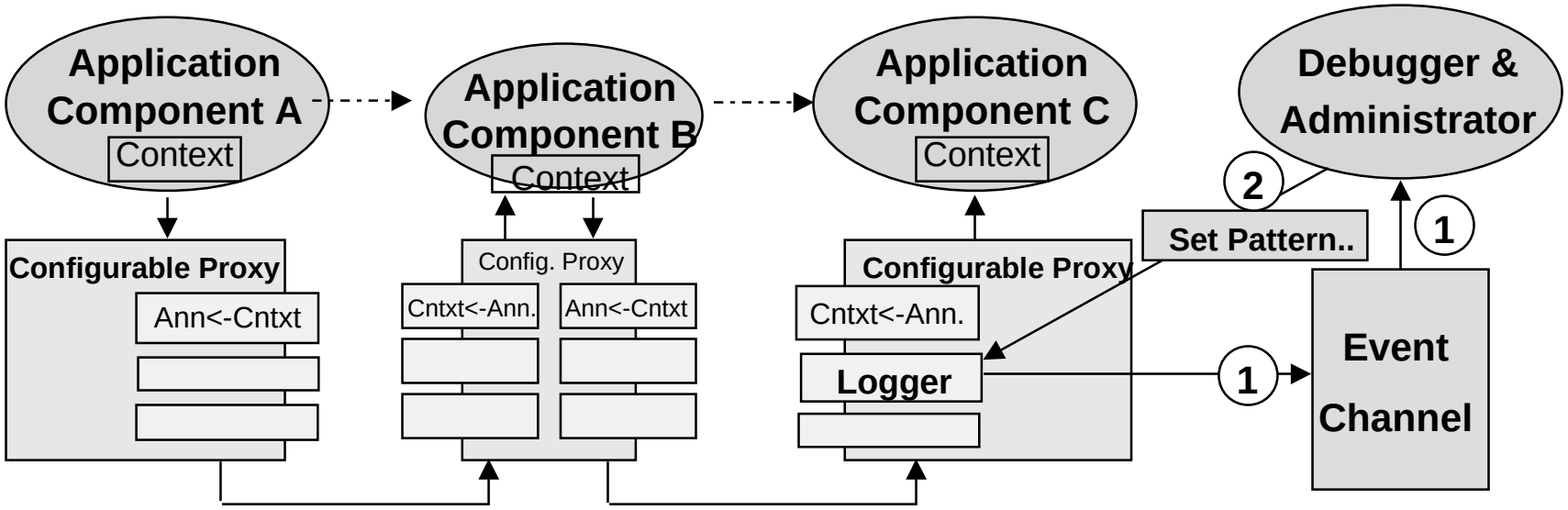
- Configurable proxies intercept communications between application components and inject services that support ilities.
- Annotated messages allow injected services to pass parameters to service peers (e.g., message priority, user-id, tracing status)
- Thread-based contexts allow annotations to pass through intermediate application nodes without requiring modification of the application code.
- The Pragma language is used to specify which service algorithms are to be injected into which communications and is largely independent of the application code

Injected Framework Features

- Futures enable asynchronous interaction between application components without the usual programming pain.
- Caching of static object attributes reduces repetitious remote requests.
- Serialization of arriving requests reduces cognitive load on application developer.
- Event channels support distribution of logging, tracing, accounting, and performance data to appropriate debuggers, intrusion detectors, and management services.
- Targets allow application components to invoke logical destinations rather than a specific object (so injected services can do load-balancing, replication, transaction processing, redirection, etc.)

Framework Usage Examples

Logging & Tracing



1. Messages arriving (or exiting) from a component are logged to an event channel if they satisfy specified criteria.
2. Logging criteria can be changed by direct calls to the interface of the logging injector.

Managing Components

- Dynamic management of components
- Components whose local functionality is (a black box) contained within the component
- Manage overall system attributes

Management

- **Performance measurement**
  - **Injectors can record time of events**
  - **Coalesced data can yield network performance information**
- **Accounting**
  - **Injectors can report billing events**
  - **Annotations can contain electronic wallets**
- **Failure analysis and recovery**
  - **Injectors can be tuned to watch for and report particular events**
  - **Injectors can remember history and respond to queries**
- **Intrusion detection**
  - **Injectors can be tuned to watch for particular events**
  - **Security agencies can take these events and recognize intrusion patterns**
- **Configuration management**
  - **Injectors can check for legal configurations**
  - **Injectors introduce their own set of configuration issues**

Event Channels and Agencies

- **Common themes in management**
  - **Publishing events to an event channel**
  - **Subscribing to the events of a channel**
  - **Causing events only on the recognition of a pattern of activities**
  - **Agencies for coalescing information**

Closing Remarks

- Currently implementing second version of framework
  - Pair of demonstration programs
    - DisDev
    - Vendoom
  - Java Visigenic/CORBA
- Controlling inter-component communication provides a management handle
  - Annotations
  - Configurable wrappers