Toward Adaptable Super Distributed Objects (SDOs): Reconfigurability in the Bio-Networking Architecture

Jun Suzuki, Ph.D.
jsuzuki@ics.uci.edu
www.ics.uci.edu/~jsuzuki/
netresearch.ics.uci.edu/bionet/
Dept. of Information and Computer Science
University of California, Irvine

Overview

• Introduction
  – Adaptability
  – Reconfiguration
  – Recap of the Bio-Networking Architecture
• Reconfiguration in the Bio-Networking Architecture
  – Reconfiguration of Network Application
  – Reconfiguration of Middleware
Adaptability

• Our focus
  – Dynamic adaptability to changes in network

• Changes in network
  – Resource availability
    • CPU cycle, memory space, disk space, network bandwidth (Ethernet, ATM, wireless, etc.)
  – Runtime application characteristics
    • Workload, user's access pattern, error pattern

Reconfigurability

• Our approach: adaptation through reconfiguration
  – Monitoring operating/network environment
    • to detect when adaptation should take place
  – Reconfiguring to adapt to changes in the environment

• Two directions
  – Network-aware reconfigurable applications
    • autonomously reconfigure their behaviors to adapt to dynamic network conditions (e.g. network load)
  – Reconfigurable middleware system
    • reconfigures their internal components to adapt to resource availability (e.g. available memory space, available transport protocols).
Bio-Networking Architecture

• Observation
  – Desirable properties of network applications (e.g. adaptability) have already been realized in various biological systems (e.g. bee colony, bird flock, etc.).

• The Bio-Networking Architecture
  – applies key biological principles and mechanisms for designing network applications.
  – a framework for developing large-scale, highly distributed, heterogeneous, and dynamic network applications.

Biological Concepts Applied

• Decentralized system organization
  – biological entities = cyber-entities (CEs)
    • the smallest component in an application

• Lifecycle
  – Each CE stores and expends energy
    • in exchange for performing service.
    • for using resources.
  – Each CE replicates itself and reproduce a child with a partner.

• Evolution
  – Dynamic reconfiguration of network applications through evolution
**Structure of Network Apps**

- **Attributes**
  - ID
  - Relationship list
  - Age
  - … etc.

- **Behaviors**
  - Communication
  - Migration
  - Replication and reproduction
  - Death
  - Resource sensing
  - State change
  - Energy exchange and storage
  - Relationship establishment
  - Social networking (discovery)

- **Body**
  - Executable code
  - Non-executable data

**Cyber-Entity’s Behavior Policy**

Each CE has its own policy for each behavior. A behavior policy consists of factors (F), weights (W), and a threshold.

- If \( \sum F_i W_i > \text{threshold} \), then migrate.

Example migration factors:

- **Migration Cost**
  - A higher migration cost (energy consumption) may discourage migration.

- **Distance to Energy Sources**
  - encourages CEs to migrate toward energy sources (e.g., users).

- **Resource Cost**
  - encourages CEs to migrate to a network node whose resource cost is cheaper.
Reconfiguration of Network Applications

• Evolution as a means to reconfigure behaviors of network applications.
  – Biological entities adjust themselves for environmental changes through species diversity and natural selection.
  – CEs evolve by
    • generating behavioral diversity among them, and
      – CEs with a variety of behavioral policies are created
        » by human developers manually, or
        » through mutation and crossover (automatically).
    • executing natural selection.
      – death from energy starvation
      – tendency to replicate/reproduce from energy abundance

Mutation and Crossover

• Weight values in each behavior policy change dynamically through mutation.
• Mutation occurs during replication and reproduction.

• Crossover occurs during reproduction.
• A child CE inherits different behaviors from different parents through crossover.
A Simulation Result

- Users (energy sources) move around network randomly.
- Evolutionary CEs gain more energy than non-evolutionary ones;
- Evolutionary CEs adapt better to dynamic network conditions.
  - by moving closer to users and avoiding network nodes whose resource cost is expensive.
  - by increasing weight values of distance-to-user and resource cost factors.

Status and Issues

- Through simulations, we have already confirmed
  - Effectiveness of energy concept
  - Effectiveness of mutation and crossover
  - Adaptability of CEs through evolutionary reconfiguration mechanisms in dynamic networks

- Issue
  - Acceleration of evolutionary process
    • by reducing energy loss and time delay.
A *Cyber-entity (CE)* is an autonomous mobile object. CEs communicate with each other using FIPA ACL. A CE context provides references to available bionet services. Bionet services are runtime services that CEs use frequently. Bionet container dispatches incoming messages to target CEs. Bionet message transport takes care of I/O, low-level messaging and concurrency. Bionet class loader loads byte code of CEs to Java VM.

**Bionet Services**

- CEs use bionet services to invoke their behaviors.
  - e.g. bionet lifecycle service when a CE replicates
- Each bionet platform provides 9 bionet services
  - Bionet Lifecycle Service
  - Bionet Relationship Management Service
  - Bionet Energy Management Service
  - Bionet Resource Sensing Service
  - Bionet CE Sensing Service
  - Bionet Pheromone Emission/Sensing Service
  - Bionet Topology Sensing Service
  - Bionet Social Networking Service
  - Bionet Migration Service
Status

• Implementation done.
  – Now in the process to document platform functionalities and improve the performance of the functionalities
  – netresearch.ics.uci.edu/bionet/resources/platform/

• Measurement work started.
  – Has confirmed bionet platform performs competitively compared with existing middleware systems and mobile agent platforms.

• The design of CEs and several other constructs is based on a preliminary version of the OMG Super Distributed Objects specification.
  – The model that SDO DSIG discussed at the DC meeting.

• Implementing evolution mechanisms that have been used and evaluated in simulation study.
  – Replication, reproduction, mutation crossover, etc.

• Will evaluate the characteristics of evolutionary reconfiguration on actual network environment.
Applications

- Content distribution
- Web service
- Peer-to-Peer networks
- Disaster response networks

Reconfiguration of Middleware

- Making not only network applications but also underlying middleware systems to be reconfigurable.

- Approach to reconfigure middleware
  - Compose middleware as a set of components.
  - Middleware
    - sense its context such as available resources and systems current configuration.
    - determine a strategy to reconfigure middleware according to the obtained context.
    - execute the determined reconfiguration strategy.
Preliminary Design Strategy

- Insert a reconfiguration layer into the bionet platform
  - Manages and controls middleware components
- Model bionet services and/or major functionalities in a bionet service as middleware components
- Manage middleware components with the Component Configurator Framework (design pattern)
**Status**

- In early design stage
  - Investigating middleware reconfiguration mechanisms using the components implemented in bionet platform.
- Designing a metaobject protocol to inspect/modify configuration of middleware components.
- MDA-like approach to reconfigure middleware?
- Biologically-inspired way to reconfigure middleware?

**Thank you**

- All the papers/documents related to the Bio-Networking Architecture are available at:
  - netresearch.ics.uci.edu/bionet/
  - netresearch.ics.uci.edu/bionet/resources/platform/
- Sponsors
  - NSF (National Science Foundation)
  - DARPA (Defense Advanced Research Program Agency)
  - AFOSR (Air Force Office of Science Research)
  - State of California (MICRO program)
  - Hitachi
  - Hitachi America
  - Novell
  - NTT (Nippon Telegraph and Telephone Corporation)
  - NTT Docomo
  - Fujitsu
  - NS Solutions Corporation