## An Overview of 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

- Motivation to and general overview of the Bio-Networking Architecture
- Design of the Bio-Networking Architecture
- Relationship between the Bio-Networking Architecture and Super Distributed Objects
- Applications of the Bio-Networking Architecture

## Motivation

- Computer network environment is seamlessly spanning locations engaged in human endeavor.
- Need a self-organizing network that supports
  - scalability in terms of # of objects and network nodes,
  - adaptability to changes in network conditions,
  - availability/survivability from massive failures and attacks,
  - simplicity to design and maintain.
- Our solution: apply biological concepts and mechanisms to network application design
  - Biological systems have overcome the above features.
    - e.g. bee colony, bird flock, fish school, etc.
- The Bio-Networking Architecture is a new framework
  - for developing large-scale, highly distributed, heterogeneous, and dynamic network applications.

## **Biological Concepts Applied**

- Decentralized system organization
  - Biological systems
    - consist of autonomous entities (e.g. bees in a bee colony)
    - no centralized (leader) entity (e.g. a leader in a bird flock)
      - Decentralization increases scalability and survivability of biological systems.
  - The Bio-Networking Architecture
    - biological entities = cyber-entities (CEs)
      - the smallest component in an application
      - provides a functional service related to the application
      - autonomous with simple behaviors
        - » replication, reproduction, migration, death, etc.
        - » makes its own behavioral decision according to its own policy
    - no centralized entity among CEs

#### • Emergence

- Biological systems
  - Useful group behavior (e.g. adaptability and survivability) emerges from autonomous local interaction of individuals with simple behaviors.
    - i.e. not by direction of a centralized (leader) entity
    - e.g. food gathering function
      - » When a bee colony needs more food, a number of bees will go to the flower patches to gather nectar.
      - » When food storage is near its capacity, only a few bees will leave the hive.

#### – The Bio-Networking Architecture

- CEs autonomously
  - sense local/nearby environment
    - » e.g. existence of neighboring CEs, existence/movement of users, workload, availability of resources (e.g. memory space), etc.
  - invoke behaviors according to the condition in a local/nearby environment
  - interacts with each other

- Lifecycle
  - Biological systems
    - Each entity strives to seek and consume food for living.
    - Some entities replicate and/or reproduce children with partners.
  - The Bio-Networking Architecture
    - Each CE stores and expends *energy* for living.
      - gains energy in exchange for providing its service to other CEs
      - expends energy for performing its behaviors, utilizing resources (e.g. CPU and memory), and invoking another CE's service.
    - Each CE replicates itself and reproduce a child with a partner.

#### Evolution

- Biological system
  - adjusts itself for environmental changes through species diversity and natural selection

#### - The Bio-Networking Architecture

- 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 (during replication and reproduction) and crossover (during reproduction)
  - executing natural selection.
    - » death from energy starvation
    - » tendency to replicate/reproduce from energy abundance

- Social networking
  - Biological systems (social systems)
    - Any two entities can be linked in a short path through relationships among entities.
      - not through any centralized entity (e.g. directory), rather in a decentralized manner.
      - six decrees of separation
  - The Bio-Networking Architecture
    - CEs are linked with each other using *relationships*.
      - A relationship contains some properties about other CEs (e.g. unique ID, name, reference, service type, etc.)
    - Relationships are used for a CE to search other CEs.
      - Search queries originate from a CE, and travel from CE to CE through relationships.
    - The *strength* of relationship is used for prioritizing different relationships in discovery.
      - A CE may change its relationship strength based on the degree of similarity between two CEs.
      - The stronger relationship is likely to lead a query to a successful discovery result.

### **CE's Structure and Behaviors**



- Attributes
  - ID
  - Relationship list
  - Age
  - ...etc.
- Body
  - Executable code
  - Non-executable data

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

## Design Strategies of the Bio-Networking Architecture

- Separate cyber-entity (CE) and Bio-Networking Platform (bionet platform),
  - Cyber-entity (CE)
    - mobile object (agent) that provides any service logic
  - Bionet platform
    - middleware system for deploying and executing cyberentities
- Design CE and bionet platform as PIMs in UML,
- Design CE and bionet platform as PSMs in CORBA IDLs and Java interfaces/classes, and
- Implement CE and bionet platform in Java

## **Design Strategies of Cyber-entity**

- CEs communicate with each other through:
  - interface CyberEntity {
    oneway void send(in string message);
    string metadata(); };
  - A subset of FIPA ACL with some extensions is used as a communication language.
    - encoded with XML
- Implemented as Java mobile code



- Each CE uses an individual thread to continuously
  - sense the nearby environment,
  - identify behaviors suitable for the current environment condition, and
  - invoke the most suitable behavior



Each CE may have 4 states during its lifetime.

It changes its state voluntarily; any CEs are not allowed to change another CE's state.

- An autonomous CE
  - receives messages from other cyber-entities.
  - continuously senses nearby environment and performs its behaviors.
  - runs on an individual thread; thus, expends energy continuously for using a thread (CPU) and memory space.
- An active CE
  - is ready for receiving messages from other cyber-entities, but
  - does not perform behaviors
  - consumes memory; thus expends energy continuously for using memory space.
- An inactive CE
  - is in "sleeping" state in which it is externalized into a file
  - does not expend energy because it does not consume any resources.

## Design Strategies of the Bio-Networking Platform

- Our approach to develop the bionet platform:
  - Identify the common networking, operating and biological functionalities required to deploy and execute CEs.
    - e.g. I/O, concurrency, messaging, network connection management, reference management, etc.
    - e.g. energy management, relationship maintenance, migration, replication, reproduction, etc.
  - Design and implement those platform functionalities as a set of reusable objects.
  - Empirically measures the platform functionalities.
- Status:
  - Design phase done, and implementation underway.
  - Measurements started.

# Architecture of the Bio-Networking Platform



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 Message Transport**

- Bionet message transport abstracts low-level operating and networking details such as I/O, concurrency, messaging, network connection management.
  - Marshaling/unmarshaling of messages issued by a CE
    - GIOP/IIOP used currently
  - TCP connection setup and management
  - Message delivery on a TCP connection
    - One-to-one messaging, currently
    - One-to-many broadcasting/multicasting (future work)
  - Threading (thread pooling) to accept incoming messages

## **Bionet Container**

- Bionet container dispatches incoming messages to target CEs.
  - Demultiplexing incoming messages
  - Dispatching incoming messages to target CEs
  - Creating CE references

## **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
    - allows a CE to change its state.
    - maintains a thread pool that contains threads assigned to autonomous CEs
    - allows a CE to replicate itself and reproduces a child CE with a partner.
    - allows a CE to reproduce a child CE with a partner
    - Mutation and crossover during replication and reproduction
  - Bionet Relationship Management Service
    - allows a CE to establish, examine, update and eliminate their relationships with other CEs.

- Bionet Energy Management Service
  - keeps track of energy level of the CEs running on a local platform.
  - allows a CE to pay energy for
    - invoking a service provided by another CE,
    - using resources, and
    - performing behaviors (i.e. invoking a bionet service).
- Bionet Resource Sensing Service
  - allows CEs to sense the type, amount and unit cost of available resources (CPU and memory).
    - CPU availability =  $N_{idle\_threads}/N_{max\_threads}$ 
      - » where N<sub>idle\_threads</sub> is the number of idle threads in a thread pool maintained by bionet lifecycle service, and
      - »  $N_{max\_threads}$  is the max number of threads in the thread pool.
    - Memory availability =  $M_{free}/M_{max}$ 
      - » where  $M_{free}$  is the amount of memory available on Java VM, and
      - »  $M_{max}$  is the max size of memory allocated to Java VM.
    - Resource unit cost is determined based on resource availability.

#### - Bionet CE Sensing Service

- allows a CE to discover other CEs running on neighboring bionet platforms reachable in N hops (platform-level discovery).
  - N = 0; discovery of local CEs running on the same platform.
  - N > 0; discovery of remote CEs running on different platforms.
- Bionet Pheromone Emission/Sensing Service
  - allows a CE to leave its pheromone (trace) on a local platform when it migrates to another platform
    - so that other CEs can find the CE at a destination platform
  - allows a CE to let other CEs know of its existence by broadcasting its metadata.
    - Other CEs may come to interact with the CE or establish a relationship with the CE.

#### - Bionet Topology Sensing Service

- allows a CE to sense the connectivity among neighboring bionet platforms reachable N hops.
  - proactive sensing
  - reactive sensing
  - hybrid sensing
  - static sensing
- Bionet Social Networking Service
  - allows a CE to search other CEs through their relationships (CE-level discovery).
  - uses 4 interfaces defined in the COS trader service – Lookup, Register, Link, Admin (not Proxy)
- Bionet Migration Service
  - allows a CE to migrate to another bionet platform.

## **Implementation Status**

- CE and its surrounding classes (e.g. relationship objects); done
- CE context; done
- Bionet message transport; done
- Bionet container; done
- Bionet Services
  - 4 services (lifecycle, relationship mgt, energy mgt., resource sensing); done
  - CE sensing; partially done
  - The other 4 services; implementation underway
- Bionet class loader
  - implementation underway

## **Preliminary Measurement (1)**



A sender CE dynamically selects a receiver CE at random, and sends an empty string message to the selected CE.

Measurements were conducted within a single machine with Java 1.4 VM, Win XP, and 1GHz Pentium 3 CPU. 64 MB heap allocated to each Java VM

- What we measured
  - Latency for message transmission between two CEs
    - How long it takes for a CE to send a message to another CE?
- Overhead sources to message transmission
  - Marshaling a message issued by a sender CE,
  - TCP connection setup,
  - message delivery on the connection,
  - message dispatching to a receiver CE, and
  - unmarshaling of an incoming message.



A single sender CE and a range of receiver CEs (1, 100, ...1000) were deployed.

Bionet platform was compared with existing distributed object platforms implemented in Java.

- Measurement results and observations
  - Bionet message transport and container are fairly efficient and comparable with existing distributed object platforms.
    - Message transmission latency was 0.17 msec when 1,000 receiver CEs were deployed.
  - Bionet message transport and container are scalable in terms of the number of receiver CEs.
    - Latency is relatively constant when the number of CEs grows, rather than it increases linearly (the average of latency was 0.179 msec.).
      - In general, increasing the # of receiver CEs increases the effort to establish TCP connections to receiver CEs (in sender side) and demultiplex/dispatch incoming messages to target CEs (in receiver side).
    - Implementation techniques such as connection sharing and hash-based demultiplexing work well.

## Preliminary Measurement (2)



A sender CE selects a receiver CE at random before a measurement, and sends empty string messages to the selected CE.

Measurements were conducted within a single machine with Java 1.4 VM, Win XP, and 1GHz Pentium 3 CPU. 64 MB heap allocated to each Java VM

- What we measured
  - throughput of a CE
    - How many messages a CE can receive and process in a second?
- The throughput is dominantly affected by
  - message demultiplexing in bionet container.



A single sender CE and a range of receiver CEs (1, 100, ...1000) were deployed.

Bionet platform was compared with existing distributed object platforms implemented in Java.

- Measurement results and observations
  - The throughput of a CE on a bionet platform is competitive with existing distributed object platforms.
    - Throughput was 2279.99 messages/sec when 1,000 receiver CEs were deployed.
  - Bionet container are scalable in terms of the number of receiver CEs.
    - Increasing the number of receiver CEs increases the demultiplexing effort. Throughput remains relatively constant as the number of receiver CEs grows (2309.27 messages/sec in average), rather than it increases linearly.
    - Implementation technique of hash-based demultiplexing works well.

## The Bio-Networking Architecture and SDO PIM/PSM

- SDO resource data model
  - The design of CE and surrounding objects is to be reflected as that of SDO resource data model
    - e.g. SDO = CE
    - SDO ID = CE ID
    - Relationship = Relationship (Organization)
    - Relationship strength = Weight of relationship
    - CE's energy level = Weight status of SDO



- The UML classes in SDO PIM were mapped CORBA IDL interfaces, structs or localized interface.
  - SDO (PIM) -> SDO interface (IDL)
  - SDO ID (PIM) -> SDO
  - Relationship (PIM)-> localized interface (IDL)
  - List of PIM constructs (PIM) -> sequence (IDL)
- A preliminary SDO PIM, which the SDO DSIG discussed and built a rough consensus on at the Orlando TC meeting, were implemented and confirmed to work well.

- SDO discovery
  - The interfaces defined in the COS trader service are used for SDO discovery
    - Discovery can be centralized or decentralized.
    - In decentralized discovery,
      - Each SDO (CE) implements Trader interfaces (Lookup, Register, Admin and Link).

### **Current Status of**

### the Bio-Networking Architecture Project

- We have been working on
  - Design and implementation of bionet platform
  - Distributed (i.e. peer-to-peer) discovery mechanism
  - Adaptation and evolution mechanism
    - genetic algorithm/operations
    - Immune system
  - Service interface description language
  - Mathematical stability analysis
- For more details, please see netresearch.ics.uci.edu/bionet/

# Applications of the Bio-Networking Architecture

- Content distribution
  - Simulations done
  - now empirical deployment underway
- Web service
  - Simulations underway
- Peer-to-Peer networks
  - Simulations done
  - now empirical deployment underway
- Disaster response networks
  - Just started

## Our Assumptions on the Disaster Net Infrastructure

- Ad-hoc networks spontaneously established in a disaster area to evacuate victims and aid emergency response crews.
- Various devises participate in the disaster adhoc nets.
  - Victims carry their own devices.
  - Emergency response crews carry and/or wear devises.
  - Emergency vehicles (e.g. fire truck, ambulance) carry devices.
  - Sensors are densely scattered over a disaster site (e.g. scattered from helicopters).

# Our Assumptions on the System Characteristics

- Large scale with a number of
  - people/organizations
  - devices
  - software objects
    - Objects represent devises, execute devise-specific functions (e.g. temperature sensing), or carry information (e.g. map, a building's floor plan and air contamination).
- Heterogeneous
  - processing, memory and networking capabilities of devises
  - functionalities of software objects
- Dynamic
  - changing network connectivity, density, and traffic
    - Connectivity and density change due to movement of users/devises and additional deployment of devises and software objects.
    - Traffic changes depending on the rescue operation stages
      - e.g. The traffic among temperature sensors increases while fire occurs.
  - intermittent availability of devises and software objects

### Goals

- To design an application architecture which
  - meets key requirements of applications running on disaster response networks (i.e. large-scale, heterogeneous and dynamic networks).
  - diminishes the maintenance/administration burden of disaster response network applications.

# The Bio-Networking Architecture and Disaster Response Networks

#### • CEs

- represents components involved in disaster response networks (e.g., victim, rescuer, service, etc.).
- provides service (e.g. temperature sensing, providing information such as building's floor plan).
- Bionet platform
  - is embedded in each devise that participates in disaster response networks.

## **Application Scenario 1: Wildfire**

- Disposable sensors are scattered over an affected area
  - e.g. temperature, wind force, oxygen, smoke sensing
  - Some of them are broken if they fall into a fire.
  - The CEs within sensors do their sensing tasks and maintain relationships with each other.

- Each fire fighter has devises (e.g. info pad, sensors).
  - The CEs within the devises may
    - direct the fire fighter to a place to extinguish a flame, even when visibility is not good, by interacting with scattered sensor CEs.
      - The CEs may suggest a safer (i.e. lower temperature, less air contaminant) route to the place from multiple options.
    - display the current positions of the fire fighter and other fighters by interacting with other fighters' CEs and the CEs that provide map information.
    - display the current area affected by fire(s) by interacting with sensor CEs.
    - sense what is happening nearby (e.g. approaching blaze) by interacting with neighboring sensor CEs, and alert the crew that.

### Application Scenario 2: Building Collapse

- The CEs within victims' devises may
  - find rescuers through passing advertisement (e.g. "I'm here" beacon) or asking its relationship partners (they will ask their partners in turn).
  - provide an evacuation path to the victim by interacting with sensor CEs.
  - obtain the first aid treatment information for injured victims by discovering and inquiring the CEs that provides the information.

- The CEs within rescuers' devises may
  - locate victims, represented by CEs, through passing advertisement or asking its relationship partners (they will ask their partners in turn).
  - display a street map or building floor plans depending on the rescuer's current position.
  - examine what is happing near the rescuer (e.g. gas leaking and approaching blaze) by discovering and inquiring nearby sensor CEs.
- A CE that provides any information may
  - adjust its population through replication, reproduction and natural selection (energy exchange) depending on the demand;
  - adjust its location through migration (e.g. toward users) and resource sensing (e.g. more CEs on the devises that provide more resources).