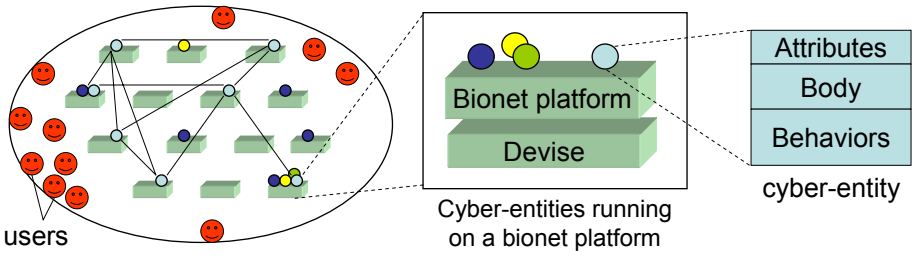
The Bio-Networking Platform:
Executive Summary

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

Implementation Strategy of the Bio-Networking Architecture

- Separate the Bio-Networking Architecture into 2 architectural components:
  - Cyber-entity (CE)
    - mobile object (agent) that provides a service(s)
  - Bionet platform
    - middleware for deploying and executing cyber-entities
      - implements a set of common functionalities among cyber-entities.
- Implement both of them in Java

Cyber-entity and Bionet Platform



- 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

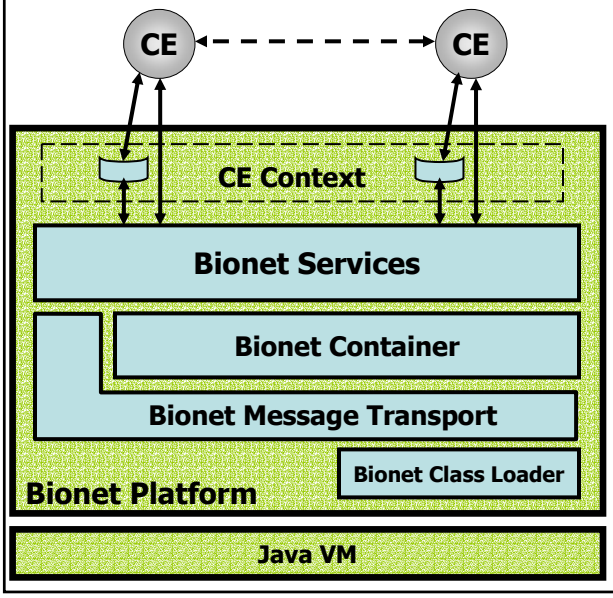
The Bio-Networking Platform

- The Bio-Networking platform (bionet platform)
  - is a middleware system that provides reusable software components for deploying and executing cyber-entities.
    - These components
      - abstract low-level operating and networking details (e.g. I/O, concurrency, messaging and network connection management) and,
      - provide a set of runtime services that CEs frequently use for performing their services and invoking their biological behaviors.

Design Strategies of the Bio-Networking 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.

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.

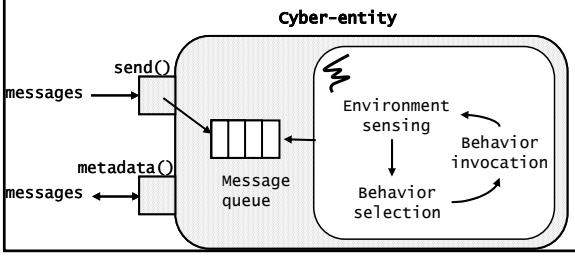
Bionet container dispatches incoming messages to target CEs.

Bionet message transport takes care of I/O, low-level messaging and concurrency.

Bionet class loader loads byte code of CEs to Java VM.

Cyber-entity (CE)

- 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

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

- Bionet services
  - is a set of runtime services that CEs frequently uses.
    - CEs uses bionet services for invoking their (biological) behaviors.
      - e.g. Bionet lifecycle service for replication/reproduction behavior
      - Bionet migration service for migration behavior
- 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

- 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 $N_{idle\_threads}$ 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.