Jack-in-the-Net: an Adaptive Networking Architecture for Service Emergence

Tomoko Itao[1], Tatsuya Suda[2], Tetsuya Nakamura[1], Miyuki Imada[1], Masato Matsuo[1]

1. NTT Network Innovation Labs
{tomoko, tetsuya, matsuo}@ma.onlab.ntt.co.jp

2. University of California, Irvine
(also, currently research professor at NTT)
suda@ics.uci.edu

Abstract

In this paper, we propose a radically new paradigm of the adaptive networking architecture for service emergence called Jack-in-the-Net (Ja-Net). In the proposed architecture, services are implemented by a collection of multiple entities called cyber-entities. These cyber-entities have functionality related to their service and follow simple behavior rules (e.g., migration, reproduction, energy exchange, mutation, death) similar to biological entities.

In Ja-Net, useful emergent behaviors (e.g., scalability, adaptation, evolution, security, survivability, and simplicity) result when individual cyber-entities interact.

1. Introduction

We envision a future where a universal network openly connects every human being and most human-made electronic objects. In addition to conventional computing devices, new devices such as sensors, wearable computers, mobile phones, vehicles, appliances, tools, and even unmanned robotic devices in space are nodes in the universal network. Thus, the universal network spans locations engaged in every human endeavor, including the home, workplace, cars, trains, airplanes, train stations, airports, and space. The universal network also supports various types of information that diverse devices collect (ranging from the temperature of the refrigerator to user personal information), creating a complex web of information.

The universal network is viewed as an extension of every single person and organization. The wide availability of open computing resources (openly shared CPU, memory, and bandwidth) and natural interfaces (e.g., speech recognition and brain-computer interfaces) allows the user to merge with the network into a unified entity, giving the person a ubiquitous presence. This unified entity (1) extends the user's identity and presence beyond his or her immediate physical surroundings, (2) extends the user's domain of control beyond his or her physical surroundings, and (3) extends the user's mental capacity beyond human memory and computational abilities.

The unified entity extends a user's identity and presence beyond his or her immediate physical surroundings by maintaining and making available to authorized parties personal information such as medical information, social security number, financial information, personal preferences, and current physical location. Such a network would, for example, monitor user's health conditions and allow doctors to instantly access personal medical information when medical attention is necessary. It would monitor user preference and allow hotel and airline reservations to be made without specifying preferences (non-smoking, etc.) every time.

The unified entity extends a user's domain of control beyond his or her immediate physical surroundings by allowing immediate control of all devices belonging to the unified entity. For example, a person could be in an airplane and activate the home DVD recorder to record an interesting program that the person just discovered from a friend.

The unified entity extends the user's mental capacity beyond normal human memory and computational abilities by utilizing any available computing resources in the network. Such computing resources provide services such as personal information management (e.g., calendar, address books), finance (banking, electronic commerce), communication (e.g., video conferencing, collaborative applications), and entertainment (e.g., multi-player games, movies).

Some aspects of this vision are already in progress today (brain-computer interface [RPR98], interplanetary networking [Was98]), but we believe a radical shift in network design paradigms is necessary to realize this vision. We believe realizing such a scenario requires a network that exhibits self-organization (requiring no central administration) with inherent support for mobility, scalability, adaptation to short and long term changes in user and network conditions, security and survivability from massive failures and attacks. Today's networks are unable to satisfactorily meet such challenges.

In order to realize the vision described above, we propose a radically new paradigm of the **Jack-in-the-Net**

(Ja-Net) [Net], an adaptive networking architecture for service emergence. Key design principles behind this new architecture are autonomous entities (called *cyber-entities*) and emergence of service created from such autonomous cyber-entities.

In the proposed Ja-Net, major components are represented by cyber-entities to reflect the fact that the universal network described earlier consists of a number of components. For instance, various devices, as well as users, on the universal network are components. Services on the universal network may consist of finer granularity service components. Each piece of information on the universal network is a component.

The cyber-entities in the proposed Ja-Net are autonomous and self-organizing. Cyber-entities also monitor the dynamically changing environment, autonomously behave and adjust their behavior and evolve to allow flexibility and adaptability of service. This is because that the scale and highly dynamic nature of the universal network prohibits human management of cyber-entities. We envision that the universal network to span wide geographical areas, supporting a large number of users and various types of devices. The universal network is highly dynamic, where user preference and usage patterns, as well as network conditions, change dynamically.

In the proposed Ja-Net, services emerge from autonomous interactions of cyber-entities. Cyber-entities autonomously form a group (called a super-entity), interact with each other to dynamically create a new service, and provide service. Desirable behaviors and characteristics emerge in the super-entity from the collective actions and interactions of its cyber-entities.

2. Service Examples

In the Jack-in-the-Net (Ja-Net), a service is implemented by a group of cyber-entities (called a *super-entity*). Each cyber-entity in the Ja-Net has service functionality related to the service and follows a set of simple behavior rules. Desirable behaviors and characteristics emerge in the super-entity from the collective actions and interactions of its cyber-entities. Some examples of services that the Ja-Net can enable include the following.

Example 1: Personalized Service

In this example, cyber-entities are assigned to users and monitor user preference and usage patterns. For instance, a user cyber-entity may monitor which hair saloon the user often makes reservations at, how frequently she makes a reservation, and what her favorite hairstyles are. When the user searches for hair saloon information on the web, it is now possible to customize the output of search based on the user profile so that the information on the user's favorite hair saloon appears first on the web. It is also possible to estimate when the user will next visit the hair saloon and which hairstyle she is interested in wearing. Consequently, a cyber-entity that provides hair saloon reservation service is dispatched to (or appears at) the user when appropriate. Also, cyber-entities that carry hairstyle catalog matching her taste is dispatched to the user.

Example 2: Dynamic Creation of Social Communities

In this example, communities (i.e., groups) of cyber-entities sharing common criteria (e.g., a community of cyber-entities providing the same type of service, providing complementary services, representing users with the same hobby) are dynamically created and destroyed, and cyber-entities dynamically join and leave such communities. Cyber-entities may, for instance, use user hand-held devices as means to move. For instance, a cyber-entity providing movie ticket sales service (in one's hand-held device) encounters a cyber-entity that provides hair saloon reservation service (in someone else's hand-held device) at a restaurant, and they may jointly start a new movie ticket sales service by offering a discount ticket to those users who make hair saloon reservations in the restaurant. The same movie ticket sales service might encounter a soccer game broadcast service at a sport bar and start displaying movie advertisement that matches the taste of users in the bar during game breaks.

Example 3: Self-Organization of Small Devices

In this example, we assume that a large number of sensor devices scattered over a wide geographical area. Assume these sensor devices support cyber-entities. Sensor devices (i.e., cyber-entities on sensor devices) monitor other sensor activities (such as what sensor data are being transmitting) and the environmental conditions (such as the density of nearby sensor devices). Sensor devices, based on monitoring results, adjust their sensing behaviors (i.e., what to sense, how often to sense, where to move to and sense, if they have movement capabilities, etc.) so that a group of sensor devices collectively self-organize and perform one sensor task.

3. Overview of the Ja-Net

3.1 Architecture Overview

In the Jack-in-the-Net (Ja-Net), a service is implemented by a group of distributed, collective entity called the super-entity. A super-entity is composed of multiple autonomous entities, each of which is called a cyber-entity (see Figure 1). Each cyber-entity has basic functionality related to their service and follows a set of simple behavior rules.

When a cyber-entity requires a service, it creates a service request and sends it to the cyber-entities within its environment. (The environment of a cyber-entity refers to the area where the cyber-entity can communicate with other cyber-entities in the area.) Cyber-entities that are members of the super-entity providing the desired service sense the request and collectively provide the requested service.

The super-entity is a generic name for any service built using the Ja-Net. The network may contain many super-entities, each with their own group of individual cyber-entities. Super-entities may be manually created. They may also emerge spontaneously through the interaction of cyber-entities. For instance, cyber-entities may create a relationship with nearby cyber-entities providing a similar or complementary service to form a super-entity.

Desirable behaviors and functionality emerge in the super-entity from the autonomous and local actions and interactions of the cyber-entities. Super-entities go through the natural selection process (to be described below) so that only successful ones survive while unsuccessful one is eliminated from the network.

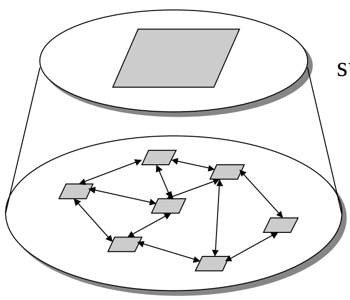

Figure 1: **Cyber-entities and a super-entity**

In addition to emergence based on autonomous local behavior of simple cyber-entities, the Ja-Net adopts the following key concepts.

Energy as a natural selection mechanism: Each cyber-entity may store and expend energy for living. Cyber-entities may gain energy in exchange for performing a service, and they may pay energy to use network and computing resources. Thus, energy may also be viewed as a unit of exchange similar to money in economic systems. The abundance or scarcity of stored energy may affect cyber-entity behaviors and contributes to the natural selection process of the Ja-Net. For example, an abundance of stored energy is an indication of higher demand for the cyber-entity; thus the cyber-entity may be designed to favor reproduction in response to higher levels of stored energy. If the energy expenditure of a cyber-entity is not balanced by the energy units it receives from providing services to other cyber-entities (i.e., indicating lack of demand), it will not be able to pay for the network resources it needs, i.e., it dies from lack of energy or starvation. Cyber-entities with wasteful behaviors (replicating or migrating too often) will have a higher chance of dying from lack of energy.

Diversity: A sufficient degree of diversity is provided at the cyber-entity level in the Ja-Net. We design multiple cyber-entities with different behaviors to perform the same task to ensure sufficient degree of diversity. Diversity is important to ensure that the evolution process is given a large enough domain of behavioral factors and resulting services on which to perform natural selection so that the system adapts and evolves to suit a wide variety of environmental conditions.

The above concepts are implemented using relatively simple cyber-entity behaviors. Due to space limitations, only a small number of cyber-entity behaviors are listed below. Note that some cyber-entities may not have all the behaviors listed below.

Energy exchange and storage: As explained above, cyber-entities may receive and store energy for providing services to other cyber-entities. They also expend energy for using network resources.

Establishing relationships: Cyber-entities may form affiliations or groups by establishing relationships with a number of other cyber-entities. For example, cyber-entities belonging to the same super-entity may establish *super-entity* relationships with each other. Cyber-entities belonging to different super-entities may establish *friend* relationships. Cyber-entities belonging to the same community may establish *community* relationships. Cyber-entities that exist on the same node may establish *local* relationships. (Relationship may also be created based on who created or owns cyber-entities and who use cyber-entities.) Cyber-entities with relationship to each other can communicate.

Sensing the environment: The environment of a cyber-entity includes all cyber-entities that it has a relationship with (namely, cyber-entities that it can communicate with) and the network resources that it can access (such as communication links, nodes and routers, CPU and buffer resources). In the Ja-Net, cyber-entities monitor their environments. For instance, a cyber-entity may learn which cyber-entities are in the environment and what services are provided by cyber-entities in the environment. This helps a cyber-entity create a new super-entity (if there is no pre-existing super-entity) or join an existing super-entity to collectively provide new service (if there already exists a super entity). A cyber-entity may also monitor to obtain information such as user information (e.g., user preference, user behavior/usage patterns) and network resource information (e.g., network topology, communication link bandwidth, CPU processing power of network nodes). Note that not all the cyber-entities may exist in the environment; for a given cyber-entity, its environment only includes those cyber-entities that it has a relationship with. Note also that environments are different for different cyber-entities.

Interaction: Each cyber-entity autonomously interacts with "appropriate" cyber-entities in the environment to collectively provide service. There are different algorithms and policies on how to find "appropriate" cyber-entity to interact with. For instance, a cyber-entity may select a cyber-entity that it has most recently interacted with.

Migration: Cyber-entities may migrate from network node to network node.

Replication and reproduction: Cyber-entities may make a copy of themselves (replication). Two parent cyber-entities may reproduce and create a child cyber-entity (reproduction), possibly with mutation and crossover changes in the child's behavior and relationship with other cyber-entities.

Death: Cyber-entities may die because of, for instance, old age or lack of energy.

3.2 Emergent Behaviors and Characteristics of the Ja-Net

Cyber-entities establish relationships (for instance, super-entity relationship), interact with each other, and form a super-entity. The super-entity is not a particular process or entity at a specific network location; rather, it refers to the emergent behavior based on autonomous behaviors, interactions, relationships of cyber-entities. The network may contain many super-entities, each with its own group of cyber-entities. Services and applications built using the Jack-in-the-Net (Ja-Net) will share a common set of important characteristics, namely scalability, adaptability, evolution, simplicity, security, and survivability. The way such characteristics emerge from the relatively simple actions of the cyber-entities are described below:

Scalability

A super-entity is scalable because all of its cyber-entities are designed to act autonomously and locally based on local information in their environments. There is no entity controlling all entities, creating no bottlenecks around a master cyber-entity when the number of cyber-entities grows. Cyber-entities repeat the same local actions and interactions in their environments even when the size and population of the super-entity increases.

Adaptation

The super-entity adapts to heterogeneous and dynamic network conditions through the emergent behavior and relationship of its cyber-entities. For instance, cyber-entities can be designed to migrate toward the source of requests when the resource cost at the migration destination is less than a given threshold, while avoiding areas of the network where resource costs are high. Cyber-entities can be, at the same time, designed to replicate when demand is high (i.e., when its stored energy level is high), and to die when demand is low (i.e., when its stored energy level reaches zero). In addition, cyber-entities can also be designed to establish super-entity relationships with a cyber-entity that they frequently interact with so that super-entities (i.e., new services) are dynamically created. The emergent result of these simple individual behaviors and relationship establishments is that the super-entity adapts its population and configuration to the amount of service requests, the source of service requests, the cost of network resources, and frequency of interaction.

Evolution

Evolution of services provided on the Ja-Net occurs through the following mechanisms.

Diversity: Diverse services emerge through dynamically creating and modifying super-entity relationships (i.e., through dynamically adding and removing cyber-entities to and from a super-entity). Diverse behaviors of cyber-entities (created through their replication/reproduction with mutations and crossovers to behavior algorithms) also contribute to the diversity of the service provided on the Ja-Net.

Natural selection: Death from lack of energy and aging steers the cyber-entity population towards more effective services, behaviors and relationships, and away from unfit ones.

Super-entity relationship may be dynamically created and modified based on the result of environment sensing by cyber-entities. There are two approaches for a cyber-entity to create new relationship:

(1) Cyber-entity monitors their environments, learn which cyber-entities to interact with, and create a super-entity relationship with such cyber-entities.

(2) Cyber-entities develop new behavioral algorithms to create relationships through mutation and crossover.

Because of the mutation and crossover processes that occur during replication and reproduction, a super-entity will have a population of cyber-entities with diverse behavioral and relationship patterns. The human designer may also introduce algorithmic diversity so that the Ja-Net can evolve in a shorter period of time.

Diversity of behavior promotes successful evolution through a large domain of behaviors [Smi86][KI98]. We expect that diversity in cyber-entity behavior coupled with variations in conditions in different parts of the network will result in the phenomenon of *localization*. Localization causes some cyber-entities in a super-entity to evolve behaviors suitable for their local environmental conditions, while other cyber-entities in the same super-entity will evolve behaviors more suitable for their own local environment.

Simplicity

The construction of the super-entity is simplified because only the relatively simple behaviors of the individual cyber-entities need to be designed, and cyber-entities can autonomously learn super-entity relationship. The other key features of scalability, adaptability, security, and survivability do not have to be directly designed. These key features naturally emerge from the simple behaviors of the cyber-entities.

Security and Survivability

The super-entity has a wide variety of emergent security and survivability behaviors and characteristics, which can be used, in addition to traditional security techniques, as extra layers of defense against attacks and failures. These behaviors and characteristics are:

Replication: Because a service (i.e. a super-entity) consists of multiple, replicated cyber-entities, even if a percentage of the cyber-entities are destroyed, the surviving cyber-entities can still carry out the function of the service or application. The surviving cyber-entities will also autonomously replicate to re-establish their initial population.

Algorithmic diversity: The code that implements the cyber-entity behaviors and relationship establishment may be implemented with different algorithms. Since attacks and failures sometimes depend on an exact algorithmic sequence or parameter, some cyber-entities may be unaffected by a particular attack or failure.

4. Design

To show the feasibility of the Jack-in-the-Net (Ja-Net), we will describe some possible designs for various aspects of the Ja-Net.

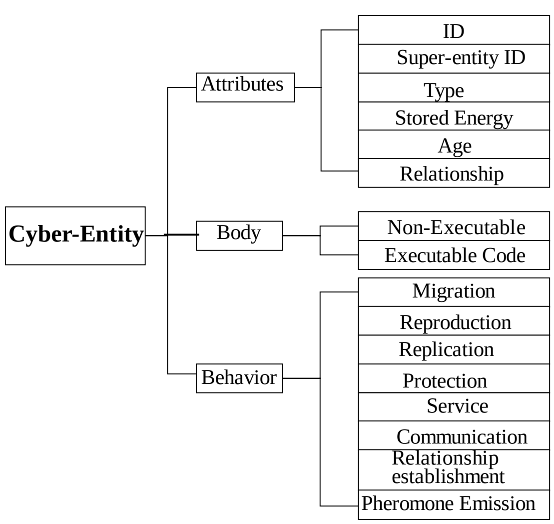

Figure 2: **Cyber-entity Components**

4.1 Cyber-entity Design

In the following, we describe a possible design for cyber-entities.

4.1.1 Structure

Cyber-entity consists of three main parts: attributes, body, and behaviors. (See Figure 2.)

Attributes carry information regarding the cyber-entity (e.g., cyber-entity ID, relationship, cyber-entity type, stored energy level, and age.). Relationship carries information regarding how the cyber-entity is related to those that are in its environment, namely, relationship type (e.g., super-entity, community, local), strength of the relationship, information about the cyber-entities that it has relationship with (e.g. their IDs, locations, service types). The strength of relationship dynamically changes as a cyber-entity learns the significance of the relationship through interactions. If a cyber-entity knows the ID and location of a cyber-entity that it has relationship with, it can directly interact with it (i.e. through unicast), otherwise it broadcasts its message to all or a subset of cyber-entities that it has relationship with.

The *body* contains materials relevant to the service that the cyber-entity provides. The body may contain, for instance, data or user profiles (non-executable data), or application code (executable code). The body of resource cyber-entities contains parameters relevant to the resources it manages, such as network bandwidth, CPU time, and memory.

Cyber-entity *behaviors* are described in the next subsection.

4.1.2 Behaviors

Behavior defines and controls the autonomous actions of the cyber-entities. Due to space limitations, we will only give a few examples of the behaviors related to evolution mechanism.

Migration

The migration behavior involves determining where to migrate to and when to migrate, i.e., considering the cost/benefit tradeoff of migrating towards an adjacent node. The benefit of moving to the adjacent node may include lower prices for resources on the adjacent node (i.e., less energy units required to use resources on the adjacent node), proximity to an energy source or a service requester, lower delays and increased fairness for all requesting cyber-entities. Another benefit of migration may include acquisition of new relationships. A cyber-entity may meet new cyber-entities at the node that it migrates to, and this increases the possibility of establishing new relationships, and thus, creating new services.

Cost may include the energy units required to use the network resources for migration and higher prices for resources on the adjacent node. Some possible factors that may impact cyber-entity's migration behavior are: move toward an energy source (i.e., a cyber-entity sending service requests), toward cheaper resources, and toward a cyber-entity with which relationship is strong. One example algorithm for controlling migration is that if the benefit of migrating to a particular node exceeds the migration cost above a certain threshold value, then the cyber-entity will decide to migrate to that node.

Replication

The replication behavior produces a copy of a cyber-entity. Some possible factors affecting the replication behavior are: stored energy level, replication resources cost, the population density of entities, and etc. A newly replicated cyber-entity acquires energy from its parent. The newly replicated cyber-entity also derives its behavior from its parent, with or without mutation. In mutation, a new set of behaviors and relationships are derived from the parent's behaviors.

Reproduction

The reproduction behavior produces a child cyber-entity from two parent cyber-entities. The reproduction behavior involves four steps: decision to reproduce, mate selection, crossover (with possible mutation), and birth.

The decision to reproduce may be affected by the factors such as stored energy, and resource cost.

The *mate selection* step seeks a partner of similar stature in the environment. Criteria for a desirable partner may include compatibility, high stored energy level, and relatively advanced age. Compatibility may mean reproduction is only allowed between cyber-entities of the same type and same software version. Higher stored energy and relatively advanced age may be desirable because they imply that the potential mate cyber-entity has suitable behaviors and relationships, and has survived a long time.

The *crossover (with possible mutation)* step mixes the behavioral rules and list of relationships of the parents to produce new behavior for the offspring. The crossover procedure may also include mutation, as in replication.

The *birth* step creates the new offspring cyber-entity. The new cyber-entity receives a new identifier, an initial amount of stored energy from both of its parents, behaviors, and relationships produced from the previous step. The content of the new cyber-entity's body is application specific.

4.2 Providing Services

In the Ja-Net, services are provided through interactions between cyber-entities in a super-entity. A cyber-entity that requires a service (including a cyber-entity that represent a user) creates a service request and sends it to the cyber-entities in its environment (i.e., cyber-entities that it has relationship with) in the hope that some of them belong to the super-entity that can provide the requested service. If none of the super-entities can provide the desired service, each cyber-entity in turn recursively query other cyber-entities that it has relationship with. Alternatively, some of the cyber-entities that are in the requesting cyber-entity's environment may self-organize and form a new super-entity to provide the requested service. Note that the process of requesting, finding, providing and receiving service is based on cyber-entity interactions.

4.2.1 Requesting Services

In the Ja-Net, a cyber-entity may request service either explicitly or implicitly.

In ***the explicit mode***, a cyber-entity explicitly issues a request for a specific service. In this mode, a requesting cyber-entity knows in advance which services (i.e. super-entities) are available and which super-entities actually provide the desired service. For instance, a user (or a cyber-entity representing a user) can query a search engine to find a super-entity providing the desired service and obtain addresses and IDs of cyber-entities that belong to the super-entity. Then, the user (or a cyber-entity representing a user) directly issues a service request to those cyber-entities.

In ***the implicit mode***, a cyber-entity does not issue a service request. In stead, a cyber-entity behavior is monitored, and cyber-entity's certain behavior or state triggers service. For instance, behavior of a user and his/her personal information may be monitored (through imbedded sensors, for instance), and his/her certain state or unconscious behavior triggers the service. The implicit mode is effective especially when users do not know which service to request, nor they are capable of creating service requests themselves.

In order to achieve effective monitoring, a cyber-entity may actively advertise the information regarding itself (e.g., behavioral and personal information of the user that it represents) to its environment (i.e. to cyber-entities that it has relationship with). Alternatively, it may make such information publicly accessible so that other cyber-entities can monitor the information.

In the implicit mode, cyber-entities (or users) do not issue service requests; rather the service is automatically provided to cyber-entities (or users), and cyber-entities (or users) select preferred service(s) from the services anonymously provided. For instance, by monitoring behavioral/personal information such as frequency of access to web pages providing certain types of audio/visual service, as well as mental conditions (e.g. getting nerves, represented by high degree of perspiration) and physical conditions (e.g., blood pressure of low: 120, high: 170), the user may automatically receive latest information about new audio/visual products, healing music being plaid as BGM, and dinner menu for blood pressure control. The capability of handling the implicit service request is one of the outstanding features of the Ja-Net.

4.2.2 Interactions between Cyber-Entities to Provide Service

In providing services in the Ja-Net, cyber-entities interact with each other in either a closed or open mode.

In ***the closed mode of interactions***, a cyber-entity interacts with only a specified cyber-entity or a super-entity (or specified cyber-entities or super-entities). The target entities of interaction are uniquely specified with their IDs and locations, and an explicit service request is issued to such entities. In this mode of interaction, among the cyber-entities that have a relationship with the requesting cyber-entity, a service request is sent only to the specified ones.

In ***the open mode of interactions,*** a cyber-entity sends a service request to all cyber-entities in its environment. All cyber-entities that are in the environment of the requesting cyber-entity receive the request, interpret the request, and only those which can process the request respond; other cyber-entities ignore the request. The service that the requesting cyber-entity will actually receive is determined by which cyber-entities are in the environment and which of them will actually respond.

In the closed interactions, the requesting cyber-entity explicitly specifies a service (super-entity), and this implies that the super-entity that provides the requested service is static. In the open interactions, any cyber-entities that are in the environment of the requesting cyber-entity can respond, and thus, the super-entity that provides the requested service is dynamic and open. In the open mode of interactions, there is no guarantee that a cyber-entity can receive the desired service, unlike the closed mode of interactions, where a cyber-entity specifies the super-entity that provides a desired service.

In both closed and open interactions, multiple cyber-entities (or super-entities) may respond to a service request. Different policies are possible to decide which to select from multiple responses. For instance, a selection policy may be "select the cyber-entities that are most recently used". Another example policy is "select those with high access frequency." Cyber-entities can also dynamically make selection decisions by sensing its environment (such as the density of cyber-entities of the same type, cyber-entities with a high energy level) or by learning successful relationships as they interact with other cyber-entities.

4.2.3 Receiving Services

Upon receiving the service, a cyber-entity may record the name of the super-entity (like bookmarks) or establish a strong relationship with a cyber-entity that is a member of that super-entity to facilitate the access to the service at a later time.

4.3. Evolution of Services

Services evolve in the Ja-Net through the creation of diversity (i.e. service emergence) and natural selection. Diverse services emerge through dynamically creating and modifying super-entity relationships (i.e., through dynamically adding and removing cyber-entities to and from a super-entity). Diverse behaviors of cyber-entities (created through their replication/reproduction with mutations and crossovers to behavior algorithms) also contribute to the diversity of the service. Note that super-entity diversity is created from the cyber-entity level diversity, and there is no super-entity level mechanism for creating diversity in the Ja-Net. Natural selection on the Ja-Net is also at the cyber-entity level. Evolution of services promotes the adaptability of the system.

4.3.1 Service Emergence

Service emergence refers to dynamic changes in the super entity membership (i.e., which cyber-entities belong to a given super entity). A super-entity may acquire a new or lose a current cyber-entity (or cyber-entities) through establishing or deleting super-entity relationship with cyber-entities. There are two mechanisms to achieve this:

Environment learning: Cyber-entities migrate autonomously. They may establish a new relationship with cyber-entities that exist on the node that they migrated to, and as a result, a super-entity may dynamically acquire new cyber-entities. Alternatively, cyber-entities may find other cyber-entities to establish new relationship with through the open mode of interactions. For instance, a cyber-entity migrates to a user's cell phone, moves around with the user, meets new cyber-entities and makes new super-entity relationship (see service example 2 in Section 2). Yet, another way for cyber-entities to acquire new relationship is through the exchange of relationship information between cyber-entities. For instance, when cyber-entities of the same service type meet, they may exchange relationships information that each has.

Genetic operations: When a cyber-entity replicates, or when cyber-entities of the same service type meet and reproduce, the relationship that the involved cyber-entity or cyber-entities have can go through genetic operations (such as mutations and crossovers) and result in new relationship.

In the environmental learning, cyber-entities change relationships based on its environment and its interactions with other cyber-entities. Since the cyber-entity environment reflects the trends of social activities of human users and cyber-entities, resulting emergent services will well reflect such trends. In the genetic operations, it may be possible that completely new and unexpected services (i.e. services that are completely different from any other existing services) emerge because of genetic changes.

4.3.2 Natural Selection

In the Ja-Net, the natural selection mechanism is based on energy exchange at the cyber-entity level.

An abundance of stored energy is an indication of higher demand for the cyber-entity; thus the cyber-entity may be designed to favor reproduction in response to higher levels of stored energy. If the energy expenditure of a cyber-entity is not balanced by the energy units it receives from providing services to other cyber-entities (i.e., indicating lack of demand), it will not be able to pay for the network resources it needs, i.e., it dies from lack of energy or starvation. Cyber-entities with wasteful behaviors (replicating or migrating too often) will have a higher chance of dying from lack of energy.

A death of a member cyber-entity of a super-entity may result in termination of the service (if no other cyber-entities in the super-entity provide the same service) or may not result in service termination (for instance, if there is a copy of the cyber-entity in the super-entity).

5. Conclusions

In this paper, the authors proposed a new architecture, the Jack-in-the-Net (Ja-Net), an adaptive networking architecture with service emergence where services are created from a group of autonomous cyber-entities. The authors are currently investigating various key research issues such as design of a simulation tool for the Ja-Net, design of a platform software that manages cyber-entities, design of a cyber-entity (e.g., behavior algorithms, service code).

References

[Net] Netgroup Web Site. http://netsearch.ics.uci.edu

[NIC98] InterNIC Web Site. http://www.internic.net

[Per96] C. Perkins. "IP Mobility Support." *RFC 2002*. ftp://ftp.isi.edu/in-notes/rfc2002 .txt

[Act99] Active Software, Inc. Home Page. http://www.activedna.com/

[Dis99] Distributed.net Project Web Site. http://www.distributed.net

[Cal98] Caltech Infospheres Project. http://www.infospheres.caltech.edu/

[Poo97] Robert D. Poor. "High Density Networks: Hyphos." Slide Presentation, Personal Information Architecture Group, MIT Media Laboratory, Oct. 1997. *http://www.media.mit.edu/pia/Pubs/HyphosSlideShow/*

[She96] Timothy J. Shepard. "A Channel Access Scheme for Large Dense Packet Radio Networks." *ACM SIGCOMM'96*, August 1996.

[Smi86] J. Smith. *The Problems of Biology*. Oxford University Press, 1986.

[KI98] K. Kaneko and T. Ikegami. *Evolution of Complex Systems*. Asakura Publishing, Japan, 1998.

[Mit96] M. Mitchell. *An Introduction to Genetic Algorithms*. MIT Press, Cambridge, MA, 1996.

[CZ97] W. Cockayne and M. Zyda, *Mobile Agents*, Prentice Hall, 1997.

[IBM96] International Business Machines Corporation. "Mobile Agent Facility Specification." OMG TC Document cf/96-08-01.

[GMP+97] L. Gong, M. Mueller, H. Prafullchandra, and R. Schemers, "Going Beyond the Sandbox," *USENIX Symposium on Internet Technologies and Systems*, 1997.