

The Model and Design of Cooperative Interaction for Service Composition

Tomoko Itao^{†,††} Tetsuya Nakamura[†] Masato Matsuo[†]
Tatsuya Suda^{1,†,††} Tomonori Aoyama^{††}

[†]NTT Network Innovation Laboratories, ^{††}University of California, Irvine, ^{†††} University of Tokyo

In this paper, we describe Jack-in-the-Net (Ja-Net), a framework to achieve adaptive service composition in a large scale, open network environment. In Ja-Net, a service is implemented by a collection of cyber-entities and provided through cooperative interaction of cyber-entities. Cyber-entities communicate with each other using Ja-Net ACL (Agent Communication Language) and establish relationships with interaction partner cyber-entities through cooperation. As relationships grow, an emergent web of cyber-entities results. When interacting, a cyber-entity selects suitable interaction partner cyber-entities from cyber-entities that it has established relationships with by examining relationships to each cyber-entity. In Ja-Net, a new service is composed of when a cyber-entity establishes a relationship with a new cyber-entity.

1 Introduction

The recent advancement of computer networks and network aware ubiquitous devices [1] put forth a vision of a universal network which openly connects every human being and most human-made objects to extend ability to a universal network. For instance, new devices such as light weight sensors, wearable computers, mobile phones, vehicles, and home appliances as well as conventional computers may become components of a universal network. Software and information content are also components of a universal network. With the increasing number of components becoming a part of a universal network that dynamically changes, a universal network is required to be self-organizing with inherent support for mobility, scalability, and adaptation to short and long term changes in user and network conditions.

Jack-in-the-Net (Ja-Net) [2][3][5] is a radically new paradigm of a network with service emergence and evolution capability. In Ja-Net, major components of a universal network are represented by cyber-entities. Cyber-entities are autonomous and self-organizing. They have functionality related to their service and follow biologically inspired, simple behavior rules (e.g., migration, reproduction, energy exchange, mutation, death).

Figure 1 shows an architecture of Ja-Net. Major

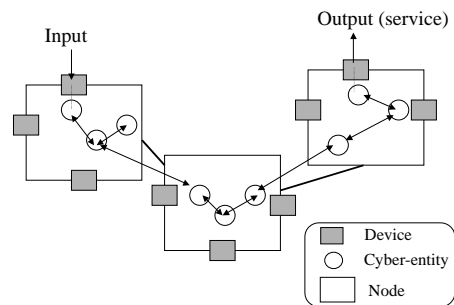


Fig. 1: Architecture of Ja-Net

architectural components include nodes, devices, and cyber-entities. Some cyber-entities may represent a device (e.g., sensor, stereo, monitor) and provide functionality to control a corresponding device, while other cyber-entities may represent a logical component (e.g., hotel reservation, coupon, MP3-encoded music file). In Ja-Net, an event in the physical world triggers cyber-entities to invoke their actions. Multiple cyber-entities interact, and a service is implemented collectively by a group of cyber-entities.

In Ja-Net, cyber-entities monitor the dynamically changing environment and behave autonomously. Desirable behaviors and structures emerge in a group of cyber-entities through interaction [4][6]. For instance, cyber-entities representing a personal identification tag, a monitor, a screen-saver, and a TV news clip may interact and provide a screen saver service and display a news clip on an idle monitor near a user. Also, Ja-Net learns about user preference and customizes the service based on user preference, resulting in an emergence of a personalized

¹A part of Prof. Suda's research conducted at UCI and presented in this paper was supported by the National Science Foundation through grants ANI-0083074 and ANI-9903427, by DARPA through Grant MDA972-99-1-0007, by AFOSR through Grant MURI F49620-00-1-0330, and by grants from the University of California MICRO Program, and Nippon Telegraph and Telephone Corporation (NTT).

news clip service.

Ja-Net has a built in natural selection mechanism for services that emerge from interactions between cyber-entities, and only successful (popular) services remain in Ja-Net, and non-successful services are eliminated from Ja-Net. In Ja-Net, services adapt and survive through emergence and natural selection of services. Ja-Net provides adaptive and survivable services through emergence and natural selection which achieve adaptability and survivability features of the system.

In this paper, we focus on service emergence of Ja-Net and describe how services spontaneously emerge from interactions of cyber-entities. The rest of the paper is organized in the following manner. In section 2, an overview of existing approaches in cooperative interaction of distributed entities are introduced. In section 3, the model of cooperative interaction for service composition is addressed. Section 4 describes the design of basic mechanisms for spontaneous service composition based on cooperative interaction. Future work is mentioned in section 5.

2 Approaches in cooperative interaction

Various frameworks exist for achieving cooperation among distributed and heterogeneous entities on a network. Hive [9] is a framework to create distributed applications. In Hive, a service is provided through multiple distributed agents where agents interact by specifying the Java interface object of the partner. Thus, interaction in Hive is limited to among those agents that mutually implement the interface object of the partner.

KQML [7][8] is a language protocol for exchanging information and knowledge. KQML defines Speech Act[16][17]-based KQML message called *performative*. A sequence of performatives is explicitly described as an Interaction protocol (IP) [14]. Agents cooperate through exchanging performatives according to an IP. Cool [10] proposes a language to describe IPs for agents to coordinate and interact. Although Cool achieves flexible cooperation between agents, IPs in Cool often are complex because it is necessary to define every possible sequence of interaction, from regular sequence to irregular sequence, in each IP to guarantee that IPs collectively process an expected task. Bee-gent [11] proposes a centralized IP model for cooperation of multiple agents to reduce complexity in IPs. In Bee-gent, a mediator agent maintains a centralized IP, and coordinates interaction of wrapper agents (i.e., wrapper of a local

process). With this centralized IP approach, Bee-gent restricts the flexibility and scalability of wrapper agent cooperation.

The Data Field (DF) architecture of ADS (Autonomous Decentralized Systems) [12] proposes information sharing facility for indirect cooperation of anonymous entities. In ADS, all entities join the Data Field (DF) where data is broadcast into the DF. Entities that join the DF receive the data and invoke a local process if possible. DF architecture is similar to blackboard architecture [13].

To achieve flexibility in cyber-entity interaction, Ja-Net adopts a data-centric interaction model similar to the DF architecture and blackboard architecture. In Ja-Net, cyber-entities communicate with each other by specifying the types of data they want to receive rather than the ID or address of a particular cyber-entity with which they communicate. In Ja-Net, cyber-entities interact using performative messages. Unlike Bee-gent, no centralized IP is used; each cyber-entity implements one or more IPs. In Ja-Net, IPs are simple unlike Cool by limiting the granularity of cyber-entities. Besides flexible interaction capability, Ja-Net is distinguished from other frameworks due to its self-organizing capability. In Ja-Net, all cyber-entities have behaviors to establish relationships one another, and strengthen relationships with distinguished partners. Organization of cyber-entities are optimized to provide desirable services for human users (See [2] and [3] for more information).

3 The model of cooperative interaction

3.1 Cyber-entity

A cyber-entity consists of attributes, body and behavior [3]. Attributes describe characteristics of a cyber-entity such as ID (CEID), service description, cyber-entity type, stored energy and age. Body corresponds to a service that an individual cyber-entity provides. For instance, some cyber-entity implements a body to control a device while some cyber-entity implements a body to make hotel reservations. In Ja-Net, to make service-related interactions simple, cyber-entity services (body) implements a unit of service at a relatively coarse level to avoid unnecessary dependency between them and maintains the communication overhead between cyber-entities at a manageable level. Behavior is non-service actions that are inherent to all cyber-entities. Examples of behavior includes: migration, relationship establish-

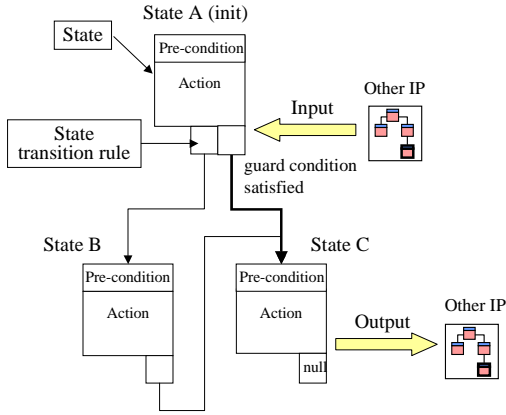


Fig. 2: Ja-Net interaction protocol

ment, mutation, cross over, etc. Body and behavior are described as an Interaction Protocol (IP) (See section 3.2).

3.2 Interaction protocol (IP)

An IP is a sequence of cyber-entity actions and message exchanges. In Ja-Net, there are two types of IPs: **Service IP** describes actions of cyber-entity services (body), and **Behavior IP** describes actions of behaviors. Both Service IP and Behavior IP are invoked as a reaction to either an external or internal trigger to the cyber-entity or message exchanges with other cyber-entities. In an IP, each interaction partner is associated with a cyber-entity role (CE role). By CE role, a set of cyber-entities satisfying specific attributes or having a specific behavior are meant.

In Ja-Net, an IP is described as a finite state model that consists of *states* and *state transition rules* (See Figure 2). A state consists of a pre-condition and an action. An action is a local process of a cyber-entity with input/output. A pre-condition is a condition on an incoming message to determine whether a cyber-entity reacts to the incoming message or not. A state transition rule for a given current state contains a next state to transit to and a guard condition or an event to trigger state transition. Examples of a guard condition include receiving a message from other cyber-entities and receiving a notification of status change of either self or other cyber-entities.

An example sequence of IP execution is given below (see also Figure 2). Upon receiving a message from another cyber-entity, the state transition rule is invoked for the current state (state A). Different state transition rules may be defined for different processing results of an action in the current state. If the guard condition of the state transition rule is sat-

Table 1: Elements of a Ja-Net ACL message

Element	Description
Performative	Speech acts
Receiver	CEID of the receiver
Sender	Self CEID
Reply-with	A tag to manage the correspondence among messages
In-reply-to	Same with Reply-with
Language	Content description language
Ontology	Content ontology
Content	Data or a message

isfied, the pre-conditions of the possible next states (state B and state C) are examined for the incoming message. The current state transmits to the next state if the pre-condition holds. Let's suppose that the state C satisfied the pre-condition. Then, an action of the current state (state C) is invoked. At the end of the action, a cyber-entity may create an output message and transmit it to interaction partner cyber-entities.

Unlike Cool, IPs in Ja-Net are simple such as a cyber-entity requesting an action from another cyber-entity, and a cyber-entity informing of another cyber-entity the result of an action. In Ja-Net, due to a relatively coarse granularity of cyber-entity services, the complexity of IP is reduced. As a result, in Ja-Net, the strict coordination facility as in the Bee-*gent* is not necessary.

3.3 Ja-Net ACL

In both Service IP and Behavior IP, cyber-entities communicate using Speech Act-based Ja-Net ACL (Agent Communication Language), which is derived from FIPA ACL [14], but specifically extended for data-centric interaction.

Table 1 shows elements of a Ja-Net ACL message. In a Ja-Net ACL, performatives such as “request”, “inform”, “query”, “not-understood” are used. For each message, an ontology may be specified. Ontology for a service domain may be defined by either developers of cyber-entity services or Ja-Net. Besides for service domains, Ja-Net defines an ontology for behavior called *ja-net-ce-behavior-ontology*, which reserves names of and defines common actions such as *advertise* and *establish-relationship*.

Data-centric interaction Ja-Net adopts in-direct cooperation model (as introduced in section 2) using Ja-Net ACL. In Ja-Net, a cyber-entity shares data with anonymous cyber-entities in its *environment* [2]. In order to interact with anonymous cyber-entities to

Table 2: Attributes of a relationship

Attribute	
CEID	M
CE-role	M
IP-name	O
Service-description	O
Energy-expense	O
Energy-gain	O
Access-count	O
Date-of-creation	M
Direction	M
Strength	M

M = Must O = Optional

provide a service, cyber-entities interpret messages that are sent from other (unspecified) cyber-entities and invoke appropriate IPs. A message may be created in an IP and broadcast in the environment, which in turn is received by a cyber-entity (cyber-entities) and triggers its (their) actions.

To achieve in-direct cooperation described above, cyber-entity interactions in Ja-Net are data-centric. In Ja-Net, a Ja-Net ACL message is represented as self-describing data where the schema of the data is explicitly described [15]. On receiving a Ja-Net ACL message, a cyber-entity parses the data based on the schema description associated with the message. The semantics of the data, on the other hand, is specific to each cyber-entity because different services have different semantics. For instance, the attribute of “date” may be interpreted as a birthday in a horoscope service, while it may be interpreted as a part of schedule in a PIM (Personal Information Management) service. By using schema description and individual semantics, cyber-entities interpret incoming messages and invoke an appropriate IP.

The in-direct cooperation with data-centric interaction assumed in Ja-Net is flexible to allow service composition (service emergence) from multiple cyber-entities. Moreover, different services may be composed of depending on which cyber-entities are present in a given environment.

3.4 Relationship

To ensure efficient service composition in Ja-Net, a cyber-entity learns useful interactions and establishes relationships with partner cyber-entities based on past experiences of successful interactions. Although a cyber-entity initially interacts with all other cyber-entities in its environment, it gradually narrows the cyber-entities to interact with based on the relationship.

In Ja-Net, a relationship contains information about a partner cyber-entity. Table 2 shows the attributes of a relationship. Strength evaluates utility of the partner cyber-entity and used to help cyber-entities to self-organize. (Self-organization in Ja-Net is out of the scope of this paper. See [2] for more information).

Relationship establishment A relationship may be established between two cyber-entities through interaction. Depending on types of messages received during interaction, there are two methods to establish relationships as described below.

Advertisement: Cyber-entities exchange advertisement messages using an “Advertise” Behavior IP. Upon receiving an advertisement message of another cyber-entity, a cyber-entity invokes “Relationship establishment” Behavior IP and creates a new relationship record. An appropriate IP and CE role of a sender cyber-entity are selected based on the information in the advertisement message and stored in a relationship record. The cyber-entity also records the CEID of the sender and other relevant attributes in a relationship record.

Service: Upon receiving a Service IP message, a cyber-entity selects an appropriate Service IP and the CE role of a sender cyber-entity. At this moment, the cyber-entity invokes “Relationship establishment” Behavior IP. Given the name of selected Service IP and CE role, “Relationship establishment” Behavior IP creates a new relationship and records the CEID, the CE role, and other relevant information that is obtained from the message. Some relationship attributes such as access-count and energy-expense/gain may be recorded in a relationship on an observation basis.

Partner selection A cyber-entity determines which cyber-entity to receive an outgoing message by examining relationships. To retrieve CEID of receiver cyber-entities, a cyber-entity firstly examine relationships that contain a CE role of the partner cyber-entity which is specified in a currently executing IP. The cyber-entity, then, identifies the CEID for such relationships. Along with CE-role, other attributes such as access-count and energy-gain may be used as keys to obtain above relationships.

4 Basic mechanisms of cooperative interaction

4.1 The system architecture

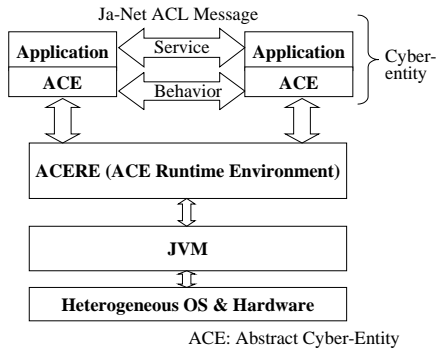


Fig. 3: System architecture

Figure 3 shows an overview of the system architecture of Ja-Net. Application part of a cyber-entity corresponds to the body of a cyber-entity. Body is implemented as a Service IP in each cyber-entity. Abstract Cyber-Entity (ACE) is a base class of a cyber-entity where all cyber-entities inherit the ACE class. Cyber-entity attributes and Behavior IPs are implemented in the ACE class. A message layer concept is introduced for message exchange between two cyber-entities where service messages and behavior messages are exchanged at service sublayer and behavior sublayer, respectively. ACE Runtime Environment (ACERE) provides a communication facility and other core functionalities for cyber-entity interaction, such as cyber-entity directory and energy management.

4.2 The internal collaboration of ACE

Figure 4 shows the internal structure of ACE. In implementing ACE, a state model is defined for each Service IP and Behavior IP (i.e., “Relationship establishment” and “Advertise”). A state model of a Service IP is created and registered with ACE for each instance of a cyber-entity. On the contrary, state models for Behavior IPs are pre-defined in ACE and inherent to all instances of cyber-entities. An IP object in ACE is a generic entity to control the execution of a given state model. An IP dispatcher receives an incoming message from ACE communication service in ACERE and determines which IP (state model) to invoke. Then the IP dispatcher invokes an IP object for a selected state model.

A relationship is established either by receiving an advertisement message of a partner cyber-entity at behavior sublayer, or during interaction at service sublayer. The following describes collaboration of cyber-entity components to establish a relationship through interaction at service sublayer.

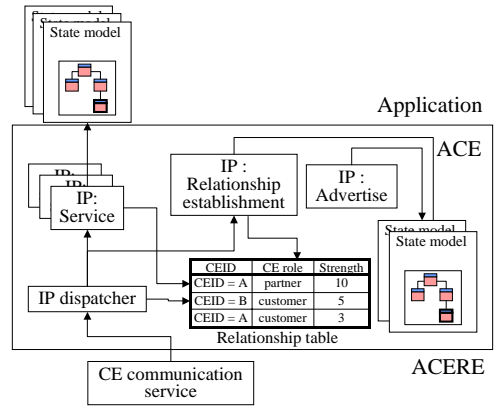


Fig. 4: The structure of ACE (Abstract Cyber-Entity)

1. Upon receiving a message at service sublayer, an IP dispatcher selects an appropriate state model of a Service IP. A CE role is assigned to a sender cyber-entity.
2. The IP dispatcher invokes a “Relationship establishment” Behavior IP by specifying the received message and the names of a Service IP and the CE role of a sender cyber-entity.
3. When the “Relationship establishment” Behavior IP terminates, the Service IP, which is selected in step 1, is invoked.

4.3 Service composition sequence

Figure 5 shows an example of message exchange sequence for providing a service, where cyber-entities representing a ticket sales (Ticket CE), coupon distribution (Coupon CE), and a user’s PDA (PDA CE) interact and provide a discount ticket sales service. Each cyber-entity implements different Service IPs and exchanges Ja-Net ACL messages. The Ticket CE invokes “Advertise” Behavior IP, when the access to its service exceeds a pre-defined threshold value, and sends an “inform” message (indicated as (1) Inform(advertise) in Figure 5).

Upon receiving an “inform” message from the Ticket CE, the Coupon CE examines the pre-condition of all possible Service IPs (service). Suppose that following is a pre-condition of “Make-partner” IP,

```
if (acl.performative == ‘inform’
    && acl.subject == ‘advertise’)
    invoke IP(‘Make-partner’);
```

and a “Make-partner” Service IP is invoked because the incoming message satisfies the pre-condition of “Make-Partner” IP. Then, the “Make-partner” IP

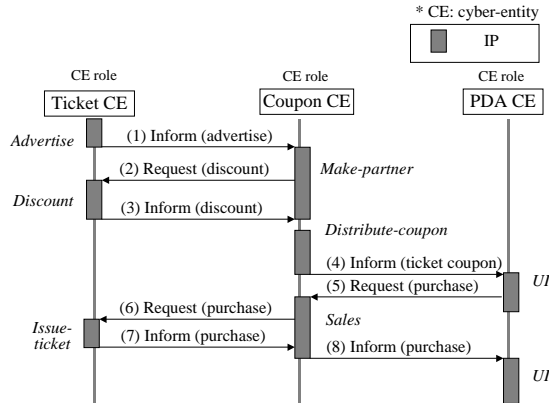


Fig. 5: Example of service provision sequence

sends a “request” message to ask for discount sales, which is accepted through “inform” from Ticket CE (indicated as (2) Request(discount) and (3) Inform (discount) respectively in Figure 5). Relationship establishment IP is invoked by the Ticket CE and the Coupon CE when receiving messages (1),(2) and (3) respectively. Consequently, the Ticket CE and the Coupon CE establish a relationship. Triggered by the establishment of a new relationship, the Coupon CE invokes the “Distribute-coupon” IP and sends a coupon information to a PDA CE (indicated as (4) Inform(ticket coupon) in Figure 5). On receiving the coupon, the PDA CE invokes an UI (User Interface) IP and displays the coupon on the screen. The PDA CE interacts with a human user, who issues a purchase request for the ticket using the coupon. The PDA CE sends a user purchase request to the Coupon CE (indicated as (5) Request(purchase) in Figure 5). This request in turn is forwarded to the Ticket CE (indicated as (6) Request(purchase) in Figure 5). The certificate of a purchase is sent back from the Ticket CE to the PDA CE via Coupon CE (indicated as (7) Inform(purchase) and (8) Inform(purchase) respectively in Figure 5). During this interaction, a new relationship is established between the PDA CE and the Coupon CE.

Once relationships are established, the Coupon CE may selectively issue a coupon of a ticket provided by the Ticket CE, and send a coupon to the PDA CE.

5 Future work

We are currently implementing basic mechanisms for cooperative interaction described in this paper. Issues that we are currently investigating include how to represent pre-conditions and what language to use to implement self describing data in Ja-Net ACL. We

are considering using XML [15] for describing self-describing data. We are also investigating algorithms for relationship establishment and partner selection, and various algorithms will be empirically evaluated for their efficiency in service composition and emergence. To realize the self-organizing capability of cyber-entities, we are investigating mechanisms to achieve strength evaluation and natural selection.

Reference

- [1] Mark Weiser, “The Computer for the Twenty-First Century,” *Scientific American*, pp.94–101, Sep. 1991.
- [2] Tatsuya Suda , Tomoko Itao, Tetsuya Nakamura , and Masato Matsuo, “A Network for Service Evolution and Emergence: Jack-in-the-Net,” *Journal of IEICEJ, B* , Vol. J84-B, No.3, pp. 310-320, 2001.
- [3] Tomoko Itao, Tatsuya Suda, Tetsuya Nakamura, Masato Matsuo, “Adaptive Networking Service Architecture for Service Emergence,” *Symposium on Applications and the Internet Workshops (SAINT)*,Jan., 2001.
- [4] K. Kaneko and T. Ikegami, “Evolution of Complex Systems,” Asakura Publishing, Japan, 1998.
- [5] The BNA project web site, <http://netresearch.ics.uci.edu/bionet>
- [6] Michael Wang and Tetsuya Suda, “The Bio-Networking Architecture: A Biologically Inspired Approach to the Design of Scalable, Adaptive, and Survivable/Available Network Applications,” *Symposium on Applications and the Internet (SAINT)*, pp.43–53, Jan., 2001.
- [7] T. Finin and G. Wiederhold, “An overview of KQML: A knowledge query and manipulation language,” Dept. of Computer Science, Stanford University, 1993.
- [8] Yannis Labrou, “Semantics for an Agent Communication Language,” Ph.D. Thesis, UMBC, 1996.
- [9] Nelson Minar, Matthew Gray, Oliver Roup, Raffi Krikorian, and Pattie Maes, “Hive: Distributed Agents for Networking Things,” *Proc. of ASA/MA*, 1999.
- [10] Mihai Barbuceanu and Mark S. Fox, “COOL: A Language for Describing Coordination in Multi Agent Systems,” *Proc. of the First International Conference on Multi-Agent Systems*, 1995.
- [11] Takahiro Kawamura, Yasuyuki Tahara, Tetsuo Hasegawa, Akihiko Ohsuga, and Shinichi Honiden, “Bee-gent : Bonding and Encapsulation Enhancement Agent Framework for Development of Distributed Systems,” *Journal of IEICEJ, D-I* , Vol. J82-D-I, No.9, 1999.
- [12] Kinji Mori, “Autonomous Decentralized Systems: Concept, Data Field Architecture and Future Trends,” *Proc. of ISADS’93*, 1993.
- [13] Barbara Hayes-Roth, “ A Blackboard Architecture for Control,” *Artificial Intelligence*, 1985, 26.
- [14] FIPA-ACL web site, <http://www.fipa.org>
- [15] XML web site, <http://www.xml.org>
- [16] Austin J. L., “How to Do Things with Words,” Clarendon Press 1962.
- [17] Searle J. R., “Speech Acts,” Cambridge University Press, 1969.