

## Client and Agent Mobility Management

Yao-Nan Lien

Department of Computer Science, National Chengchi University  
Taipei, Taiwan, R.O.C.

### Abstract

To make information ubiquitously available to the people in the world requires not only the Information Superhighway, but also a non-traditional computing paradigm, such as the intelligent messaging, to overcome the intermittent connection problem inherent in a mobile environment. This paper studies the "client" and "agent" mobility problems within a ubiquitous information service environment. The client mobility allows a client to change his/her location across different logical networks. The agent mobility allows an intelligent message to travel in a service network. Both mobilities require extensive operation, administration, and maintenance (OA&M) supports. This paper proposes a hybrid infrastructure that can incorporate both centralized and distributed supporting facilities to offer an effective and economic OA&M support to both mobilities. This infrastructure also makes use of personal Internet facilities in the mobility management to share the expensive OA&M cost. Many issues related to the mobility management can be solved more economically by using personal Internet facilities.

### 1. Introduction

Because of the advance of computer and communication technologies as well as the promotion of National Information Infrastructure (NII), the progress of *mobile computing* is accelerating into a revolutionary speed making the dream of ubiquitous information service a reality [3,4,14,21,24]. The goal of a ubiquitous information service network is to provide information to the users anytime anywhere. To accomplish this goal, the service network must be supported by some ubiquitously available communication networks and be able to access to various information resources. More importantly, it must have a layer of service system providing cost effective operation, administration, and maintenance (OA&M) functionalities. (For instance, a service network will not be able to survive if it doesn't have a reliable billing system.) Among many critical OA&M issues to be addressed, the mobility management is

indeed an important one. We have been seeing many researches devoted to providing various mobilities to the mobile users. The most popular mobilities are the terminal mobility providing location independence and the personal mobility providing the communication device independence to the users. However, none has been devoted to providing the mobility across logical networks, called the *client mobility*, which is the first problem this paper wants to address. With this mobility, the level of ubiquity offered by an information service network will be greatly enhanced. For example, one would be able to send an urgent voice message to another regardless where he/she is; if he/she is on a cellular phone, the network will deliver the message to that phone; if he/she is on the Internet, the message will be delivered to him/her by Email. It can also persistently alert the terminating party before the message is received.

To fit into the popular client-server model, the users can be viewed as clients and service providers can be viewed as servers. Theoretically, the problem can be greatly simplified, if we can use distributed computing technology to integrate all servers into a single logical server such that clients can access the network resources transparently by interacting with a single server. Unfortunately, applying distributed computing technology in such a scale will have to take a much longer time to realize in the real world. Thus, clients will have to access network resources in a prescriptive fashion by interacting with individual servers step by step to accomplish a complicated task. However, in most mobile computing environments, the nature of communications is intermittent and the battery energy is limited. Thus, it is very difficult to accomplish a complicated task that requires its client to interact with multiple servers intensively. A non-traditional computing paradigm, the *intelligent messaging*, that allows clients to interact with multiple servers in a dynamic fashion has been brought up to cope with this problem [7,9,10,12,13,17,25].

Simply speaking, an *intelligent message* is an electronic message that carries a computer program, whether procedural or declarative, that can be

executed by the receiving servers on behalf of the originating client. The program in the message can also instruct a receiving server to forward automatically the message itself to another server, on which the program is executed continuously in a pipeline fashion. Such a message is also known as an *intelligent agent* in other fields [7,11]. In *Telescript* terminology, an intelligent message is called an *agent* [25].

Since an active intelligent message may be traveling in a service network, the originating client may not be able to trace or control its operation directly. A service network must provide some mechanisms allowing its clients to trace and control these messages. This problem is referred to as the *agent mobility management*, which is the second problem this paper wants to address.

Both client and agent mobilities will be elaborated in the rest of this section. Various OA&M supporting models will be discussed and evaluated in Section 2. A hybrid OA&M supporting model will also be shown in the same section. The network infrastructure according to the proposed supporting model is described in Section 3. The rests of the paper illustrate some problems related to the management of both mobilities. Some solutions will also be shown under the proposed supporting infrastructure.

### 1.1 Client Mobility on Information Service Networks

The telecommunication network technology is moving toward providing more mobility and flexibility to the users. The wireless cellular technology is a typical example. Recently, Personal Communication Services (PCS) makes use of the Intelligent Network technology to push the telecommunication technology further. It provides a higher degree of mobility to the users such that a user can move across different telecommunication networks [16,18,26]. However, these mobilities are all limited to the physical networks of similar types. More specifically, they are confined within a single logical network that provides circuit connections between communicating parties. There is no infrastructure existing today that allows a cellular phone to send a message to or to receive messages from the Internet by its phone number. (A good application of this capability is to use the popular Internet-phone to call a real phone.) To avoid such a

compromise on the ubiquity objective, it is important to support the client mobility to allow the communications between logical networks. The relationship between logical and physical networks is depicted in Figure 1.

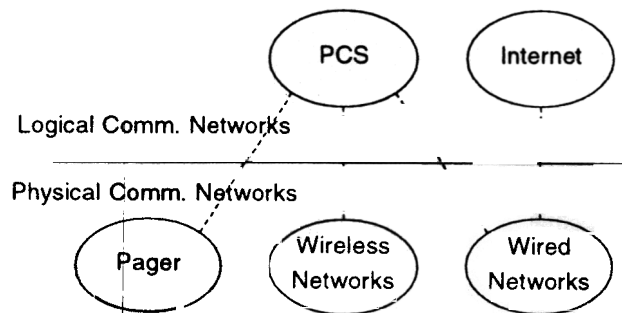


Figure 1. The relationship between logical and physical communication networks.

Various mobilities are briefly defined as follows:

- **Terminal mobility:** provides location independence to communication devices. The first generation wireless networks, such as AMPS, provide this mobility.
- **Personal mobility:** provides device independence to users. Users can use any communication device to access a network. The "follow-me" feature on a typical Public Switched Telephone Network (PSTN) is an example of personal mobility with or without the terminal mobility. Together with the terminal mobility, one can use any wireless communication device to access to or be reached from the same network. The second generation wireless networks, such as GSM, provide both terminal and personal mobilities. (Please note that the personal mobility described in many PCS literatures implies the terminal mobility, while ours don't.)
- **Physical network mobility:** provides physical network independence to users. With this mobility, a user has a unique directory number (phone number) across several communication networks. The PCS concept is to provide this mobility to the users such that users can access to or be reached from several different communication networks transparently.

- Logical network mobility (client mobility): provides logical network independence to users. A user can access to or be reached from different logical networks with a single addressing mechanism independent of the logical network where the user is being connected. As mentioned previously, there is no existing infrastructure today that provides this mobility yet.

All but the last mobility have been addressed by the telecommunication community. Their solutions can be found in various literatures [16,18,26,27]. We expect that the client mobility will not be put on the table by the telecommunication community in the near future. Therefore, the computing community will have to handle it by themselves.

## 1.2 Intelligent Messaging Paradigm and Agent Mobility

The quality of wireless communications is much more difficult to achieve than wired communications because the surrounding environment interferes with the signal, blocking single paths and introducing noise, echoes and multiple signal paths [1,5,9]. As a result, wireless connections are of lower quality than wired connections: lower bandwidth, high bandwidth variability, higher error rate, and more frequent spurious disconnection. They are also much more expensive than wired counterparts.

As we stated in Section 1.1, because of this impairment in mobile computing environments, it is very difficult to accomplish a complicated task that requires an intensive interaction between users and service networks. The intelligent messaging paradigm, that allows clients to interact with multiple servers in a dynamic fashion would be a promising approach to cope with this problem [7,12,13,14,15,17,25].

We follow the Telescript terminology to call an intelligent message as an *agent*. In many cases, an agent can be viewed as a representative of a client. It travels in a service network acting as a client to request services from the servers it visits. However, because a client can submit more than one agent to accomplish a single task, an agent may not be always equivalent to a client. A good example is shown in [15].

Following questions related to the mobility management are raised in [15]:

1. How to locate an agent in a service network?
2. How to locate a client?
3. How to know the status of an agent?
4. How to control the execution of an agent that is traveling in a service network?
5. How to trace the execution of an agent (e.g. for debugging or auditing purposes)?

The problems raised here are referred to as *agent mobility* in this paper. They are by no means exhaustive. Furthermore, there are other issues such as transaction and security supports needed to be addressed as well [15]. This paper only focuses on the agent mobility management.

To make an intelligent messaging service network commercially viable, a high quality and cost effective agent mobility management system must be in place to guarantee a certain service quality.

## 1.3 Open Service Network Architecture

An information service requires at least following elements to be successful: (1) some killer applications, (2) cost effective OA&M, and (3) adequate financial reward [15]. Traditional telecommunication networks such as PSTN and 800 Toll-Free service used to take considerable resources and long deployment duration to establish. One major resource drain in such networks is the OA&M (and provisioning in some cases). It will be impractical to demand the comparable resources to support the OA&M functionalities in many perspective information services. As we have mentioned previously, we expect that the computing community will have to rely on themselves, rather than the telecommunication community, to develop and deploy the demanded functionalities. All infrastructures and solutions proposed in this paper are based on such an assumption that **requires no change to the existing telecommunication network**. To achieve this, we employ the open service network architecture proposed in [15], which separates service networks from transport networks. It also allows services of any scale and any quality to be added into the network easily. Service providers can choose whichever operation model and the quality level based on the resources available to them. That architecture is summarized in this subsection.

Basic entities are servers each providing a specific information service. They are connected by various logical or physical communication networks such as the Internet, the PSTN, or the ARDIS radio network. The networks that provide connectivity services between servers and clients are referred to as the *transport networks*. Any number of server can be integrated together to form a *service network* providing higher level services to their clients. A server can participate into more than one service network. A *terminal* is a physical device that allows a client to interact with a transport network (and service networks). A terminal could be a telephone, a Personal Digital Assistant (PDA), a desktop PC, or a workstation, etc. (A terminal is associated with a transport network, while a client is associated with a service network. There is no fixed relationship between a client and a terminal.) This architecture provides the transport network transparency as well as the required flexibility to the service networks. The relationship between service networks and servers is very similar to many real world systems such as travel agent networks and airlines. An airline can be viewed as a server and a travel agent network can be viewed as a service network. A travel agent integrates many airlines together to offer its customers total solutions; and an airline can give ticket selling contracts to many travel agent networks. (A complete traveling package may consist of several flights from various airlines and probably with hotels and rental cars reserved. Not only the packages are more complete, it may even be cheaper as compared to DIY solutions. Readers are referred to [15] for details.

## 2. OA&M Supporting Models

At a more abstract level, the mobility problems mentioned above are not really new to other areas such as PCS and distributed computing. Many solutions can actually be reused. However, the success of an information service relies on **whether it can be economically managed or not**, not only on the technical merit. Therefore, the most important issue to address before finding any solution is **to find an infrastructure on which a cost effective OA&M can be easily deployed**. In this section, a hybrid operation model that integrates centralized and distributed OA&M together and makes use of **personal Internet facilities** is proposed to reduce the

operational expense significantly without a compromise on the global network integrity. Readers should keep in mind that many algorithms presented in this paper for mobility management seems not very different from what we can see in other telecommunication systems. However, the major focus of this paper is actually on the operation infrastructure itself. **It does not require any change to the transport network.** (As a counter example, most solutions proposed so far to support *mobile IP* require some changes to the base stations in wireless networks. We believe that it is not easy to convince the telecommunication community to put this request on their current agenda.)

### 2.1 Centralized vs. Distributed OA&M Supports

There are many ways to operate a service network. Among them, centralized and fully distributed supports are two extremes. Prodigy and American Online are typical examples that use centralized support; while the *World-Wide-Web* (WWW) is a typical example that uses fully distributed operation since every WWW server is completely autonomous [2]. In general, a commercial service network that guarantees a certain service quality usually uses centralized support, while a voluntary service network that doesn't guarantee any quality of service usually uses distributed support for lack of funding and liability. The open service network architecture allows service providers to choose whatever operation management models that are the most appropriate to them.

Centralized OA&M support is easier to achieve higher service quality. However, it usually needs much longer time to deploy a new service. For example, current telecommunication networks are very reliable that the average total down time of every component can be as low as three minutes per year; while it usually takes multi-million U.S. dollars and several years to develop and deploy a new feature. On the other hand, distributed support is usually more flexible in introducing new services, but has lower service quality. For instance, it is completely autonomous to introduce new services into the WWW network; while their quality of service are completely hectic. (How many broken links you have encountered so far? How do you tell a service is fake or real? How many people dare to give out their credit card numbers to place orders on electronic markets?)

## 2.2 Hybrid OA&M Support

The open service network architecture accepts both approaches. However, many perspective information services will not be able to afford the expensive centralized OA&M support, while they need certain level of service quality beyond what a distributed approach can offer. Therefore, we propose a hybrid OA&M support structure hoping to take advantages of both approaches. In this approach, all OA&M functionalities are classified into two categories: distributable and non-distributable. Distributable functionalities can be supported by any server in the network or even client's own resources. Non-distributable functionalities must be supported by designated servers. Those critical and more appropriate to be managed centrally, such as security and billing functions, are classified as non-distributable and must be managed centrally. It is yet to be researched to define and to classify all OA&M functions. We believe that many perspective services will be benefited from this hybrid OA&M supporting model.

The OA&M experiences we have so far are all centralized since most current commercial telecommunication networks and information services are all centrally managed. Therefore, the challenge is really on the distributed side due to a lack of real experience. One may think that we can emulate the mobility management infrastructure adopted in PCS. However, those solutions may not be directly applicable to the perspective service networks from business point of view. In the traditional wireless cellular networks, each mobile unit within the covered area must always be supported by a base station. Many OA&M functionalities such as *visitor location register* are carried out by these base stations. However, many service networks do not have that kind of resources to make their own base stations ubiquitously available. It is also impractical to expect the telecommunication community to modify their base stations to accommodate our needs. Thus, A better solution is to have all servers as well as users' facilities work together cooperatively using top level communication protocols, such as TCP/IP and HTTP, to manage client and agent mobilities.

Judging from the fact that the Internet users are growing in an exponential rate, we can make a reasonable assumption that most of the mobile

computing users in the future will have stationary access to the Internet from their offices or homes. These facilities are called *Home Base Nodes* (HBNs), which will be explained later in Section 3.3. We also assume that each mobile device that has the intelligent messaging capability (such as a PDA) has Internet accessing capability. An intelligent messaging user can choose to use the facilities provided by the service providers or his/her own HBN to manage the client and agent mobilities. Readers should not be difficult to figure out a number of algorithms similar to the HLR/VLR (Home Location Register and Visitor Location Register) concept to do that.

One may argue that clients' own Internet facilities are completely redundant that the same functions can be carried out by the network facilities. However, since the overhead of managing an agent is much higher than managing a cellular phone by several orders, completely relying on the network facilities will make the service much more expensive than the already too expensive cellular phone services. We believe that our infrastructure proposal is technically simple, and very practical from business perspective.

## 3. Infrastructure for Mobility Management

### 3.1 Basic Definitions

In this section, we propose an infrastructure that allows personal Internet facilities to participate in the management of client and agent mobilities. We assume a client can move across different logical communication networks, but **an agent can only travel within the Internet**. We make this assumption for practical feasibility reasons. Nevertheless, limited interface capability may be available to agents such as sending a voice message to a cellular phone.

#### 3.1.1 Entity Identification

In this architecture, an entity (server/client/terminal) may have many IDs each corresponding to a network, either service or transport network. For simplicity, we refer transport network IDs as physical IDs, and service network IDs as logical IDs.

Within a service network, we must be able to reach each entity, either a client or a server, by its

logical ID independent of the transport network it is being connected. Therefore, in a service network that supports the proposed polymorphism, a service (name server) is needed to convert a given logical ID into its physical IDs, including the transport network ID and the terminal ID. Once its physical IDs are known, the terminal location can be determined by the transport network itself. For instance, if a service network knows that a client is accessing the network via a cellular phone system, the network can reach that client directly by his/her cellular phone number without knowing his/her physical location.

### 3.1.2 Agent Identification

Every agent must have an identification that is unique with respect to a particular service network. A simple two-segment ID structure, (client-ID, message-ID), would be sufficient in many cases, where the message-ID is a unique sequence number generated by the client.

### 3.1.3 Agent Status

The following is the list of agent status observable by the external entities such as clients or other authorized entities:

*running* - an agent is being executed by a server.

Please note that the running state defined here is different from what is defined in the traditional operating systems. A running agent here may actually be blocked waiting for local resources or is waiting for an external message such as the input from its originating client. Since a blocking state is not observable by the external entities, it is still considered in the running state. However, for reference purpose, we call such an agent to be in *spinning* state.

- *hopping* - an agent is being forwarded to another server.
- *terminated* - an agent is terminated.
- *suspended* - an agent is suspended in the middle of or before an execution by an authority external to that agent.
- *frozen* - an agent in a server is not being executed, but is waiting to be forwarded to another server.

The difference between "spinning" and "suspended" is that a spinning agent can resume its execution by itself without any external permission, while a suspended agent can't. (Even if a spinning agent is waiting for something, it can always resume itself if it decides not to wait.)

### 3.1.4 Current Execution Node (CEN)

- *Current Execution Node* - the node that an agent currently resides.
- *Registered Current Execution Node* - the current execution node of an agent shown in its status holder, which will be explained later in Section 3.4.

### 3.1.5 Execution Log

The execution status of an agent such as which node it came from and which node it was forwarded to might have to be saved in each of the servers visited by that agent. This is especially useful for tracing purpose.

## 3.2 Facilities for Mobility Management

### 3.2.1 Network Management Center (NMC)

As we stated in Section 2, a central facility is needed to support all non-distributable management functions as well as distributable functions if it is needed. Typical OA&M functions are client and server registration, authentication, name server, coordination, or client specific services. Although, it looks like a single node, it may actually be a number of nodes distributed over a network. For simplicity, we assume a NMC is a single node.

In general, a commercial service network usually needs a NMC for such functionalities as billing, authentication, security, etc.

### 3.3 Home Base Node (HBN)

Even though a mobile client may change its locations from time to time, it usually has a home location and a most frequently used Internet access point such as a personal account on an Internet-connected system or a PC, called *Home Base Node*. This infrastructure proposes to make use of HBN to share the OA&M workload. One may argue that a mobile user may not necessarily have a personal Internet facility working as HBN. However, since a mobile terminal device is very vulnerable in a hostile environment, it is not wise to use a mobile device as the primary information

repository. A more reasonable working model for most users is to use their mobile computing devices only for temporary cache and having their HBNs as primary working systems. Further, when a client subscribes to a service network, his/she can choose to use or not to use his/her own HBN. A lot of overhead in managing a service network can be saved by using HBNs. A HBN can also offer auxiliary computing resources to help mobile terminals to cope with their residential resource limitation. For example, a fax sent to a client can be converted into text data in his/her HBN using Optical Character Recognition (OCR) technology first and then forwarded to the client's mobile terminal with a much smaller data size. Other example applications that a HBN can perform are: location register, fax server, answering machine, news filter, etc.

In fact, if a large portion of the workload is performed on the HBN itself, a mobile terminal can be treated as an intelligent terminal for that HBN.

### 3.4 Status Holder

The *status holder* for an agent is a place to store the status of an agent so that the client or other authorized entities can access this information easily. It can be any node such as client's HBN or the original access node, or even the NMC itself. A system may require each agent to report its status to its status holder on the designated events such as arrive-a-node, suspended, frozen, etc. A client can choose whatever the events his/she is interested when he/she submits an agent. An alternative status holder may be needed when the availability of the original status holder is a concern. A client can even request an agent **not** to report its status to save communication cost. These all depend on implementation details.

### 3.5 Dynamic IP Address

If a client is on the Internet, it may need an IP address to be reached. There are some attempts trying to offer a mobile terminal an IP address permanently regardless of its location such that a client can be reached by other Internet entities transparently [19,20]. Actually, in such a scheme, the burden of mobility management is shifted to the physical communication networks. This is a theoretically very attractive approach. However, this scheme needs a revolutionary change on the

transport networks (TCP/IP, for instance). Thus, it may have to take a long time to realize such a scheme. Fortunately, there is another cheaper alternative available today. Some Internet access servers today can assign IP address dynamically to their clients. Typical examples are Point-to-Point Protocol (PPP) and Serial Line Internet Protocol (SLIP). These protocols are quite popular today among dial-up users. With these two protocols, we can construct a ubiquitous information service network today by setting up a server with PPP/SLIP and Internet access capabilities in each major city. A client can use a cellular modem on his/her notebook running PPP/SLIP to dial-up the nearest server. Although cellular connections are very expensive, the client only needs a very short time to submit an intelligent message to the service network or to retrieve the results back. Nevertheless, this scheme is not perfect yet because it is more difficult to send a message to a client when it is not actively connected to the network. The client mobility management system must be in place to make a client reachable from a service network.

## 4. Location Support

One of the most important issues in mobility management is the ability to locate an entity such as an active agent or a client. In this section, several solutions will be proposed based on the infrastructure shown in the last section.

### 4.1 How to locate a client?

As discussed in Section 2, due to the separation of service and transport networks, the client location problem must be handled by the service networks themselves, not the transport networks. Each service network may have its own scheme to handle this problem. In the proposed hybrid OA&M infrastructure, the facilities available for mobility management are quite rich as compared to the traditional PCS approach. Both NMC and HBNs are available for keeping track of client locations. By distributing the location registration functionalities to these facilities, the traffic to the NMC can be significantly reduced, although it may take more messages and longer time to locate a client.

A straightforward scheme is very similar to the HLR/VLR scheme in PCS. Each moving client

registers his/her current location in his/her HBN or NMC. The location information should include the transport network ID and the terminal ID within that transport network, such as the (*Internet, IP address*) pair. When locating a client, an inquiry is sent to the NMC to obtain the client's HBN, then another inquiry is sent over there to obtain the registered location.

## 4.2 How to locate an agent?

After an agent is submitted into a service network, the client may need to know its current location in order to inquire its status, or to control its execution, etc. A simple way is to send another agent, called *search agent*, to search the original agent along the original path, or to send a message to every server where the agent might have visited. There are some problems associated with these straightforward solutions:

The cost of sending many wireless messages over a wireless link to locate an agent may be too high to justify. (The messages in a mobile environment are usually very expensive.)

2. The path that an agent took may be non-deterministic such that it is difficult to trace.
3. A sequential search may take too much time to locate an agent.

Therefore, better ways to locate an agent are needed. The concept of *status holder* is specially useful to support such a need, which will be shown in the following subsections.

## 4.3 Chase-from-holder Algorithms

When using a Chase-from-holder algorithm, a search agent will visit the status holder of that client first to inquire the registered location (i.e. the registered CEN) of the target agent and then route to that node. If the target agent is found in that node, the search agent sends the result back to the client and terminates. Otherwise, it uses a trace algorithm to chase the original agent along the original route. Actually, the Status Holder Independent Search Algorithms described in the following subsections can be used here if it is necessary.

## 4.4 Status Holder Independent Search Algorithms

When a status holder is not available for search, or when the agent being searched loses contact with its

status holder, a blind search is inevitable.

### 4.4.1 Binary Search Algorithms

A search algorithm similar to the binary search in searching a value in an array of data may be good for blind search. It can only work in deterministic routing where the expected visiting nodes as well as the visiting sequence are known in advance.

### 4.4.2 Execution Time Dependent Search Algorithms

If a client has a good estimation on the service time in each step, he/she might have a good guess on the current location of an agent. By using this information in a search, it may be able to reduce the search time significantly.

## 4.5 Related Issues

### 4.5.1 Exact Search vs. Approximate Search

Because the execution of an agent continues to progress, the exact current location of an agent might have already changed when the client receives the acknowledgement. To make the location of an agent remain unchanged after it is found, a freeze agent, which will be explained in Section 6, has to be submitted together with the search agent. (Don't forget to resume that agent later!) Otherwise, only an approximate location will be obtained. Nevertheless, an approximate location might already be sufficient in many cases.

### 4.5.2 Non-deterministic Execution

The search algorithms presented so far all assume that the agent to be searched traveled along a predetermined path. It may not be the case when the path it took depends on real time conditions and thus, is non-deterministic. In this case, the chase-from-holder algorithms are more appropriate. It is yet to be researched if the status holder is not available.

### 4.5.3 Lost Agent

An agent may get lost due to a failure in the agent itself, the server it resides, or the network it travels such that no search agent can find its status. Sometimes it has no harm to just ignore the lost agent. However, it might be necessary to recover the lost agent in some critical situations such as business transactions. This problem is further complicated when concurrent execution is allowed because it is

difficult to know whether all child agents are alive and all finish their assignments. Further research on this issue is needed.

#### 4.6 Concurrent Search

Sequential search may take much longer time than what a client can tolerate. A client may submit more than one search agent into a network to reduce the search time. A search agent may also create child search agents by itself to cover all possible paths in non-deterministic situations such as an if-then-else decision. (The actual route an agent took might depend on real time conditions or on the information it obtained in the course of its execution.)

Open issues are:

- How to converge forked agents?
- What to do if some child agents, or even the parent search agent itself gets lost?
- How to terminate all child agents?

#### 5. Status Inquiry

The most straightforward way to inquire the status of an agent is to send a search agent together with a status-inquiry agent. On the other hand, if an agent always reports its status to its status holder, its status can be easily obtained right from there. Again, because the execution of an agent continues to progress, the exact status of an agent is not easy to obtain due to the potential message traveling latency. One can send the status inquiry agent directly to the current execution node or let the status to be sent directly to the client (with higher communication cost) rather than its status holder if the situation is permitted. One has to balance the accuracy and the cost according to his/her own need.

#### 6. Execution Control

After an agent is submitted, a client might want to regain the control of its execution. For example, if one finds that the balance in his/her bank account is not enough to pay the bill for a submitted shopping agent, or he/she may simply change his/her mind, he/she has to terminate that agent before it is too late. The defined control functions are:

- *terminate* - kills an agent;

- *freeze* - postpones the forwarding of an agent until a resume message is received;
- *suspend* - suspends the execution of an agent until a resume message is received; and
- *resume* - resumes the execution of a suspended or frozen agent.

An agent that carries an execution control message is called a *control agent*. It may also be called by its control function if it is needed. There is an open issue: what if a frozen or suspended agent never gets resumed or terminated? Exceptional cases are always possible in real life. Therefore, some kind of garbage collection function must be provided to control these orphan agents.

#### 6.1 Look-Ahead Control

A client might want to send a control agent to a node ahead of time before the target agent arrives that node. For example:

- to change the original plan (to patch a program), or
- advanced freeze or suspension (to save searching time in a binary search, for instance).

The control function will be performed when the target agent arrives the node. The defined look-ahead control functions are:

- *terminate* - kills the execution when the target agent arrives;
- *freeze* - executes the agent when the target agent arrives, but postpones the forwarding of the agent to the next node until a resume message is received;
- *suspend* - suspends the execution of the target agent immediately when it arrives at the node until a resume message is received;
- *cancel* - cancels a previously submitted control request.

There are some exceptions: a target agent might never reach the node such that the look-ahead control agent might not get a chance to apply and become an orphan eventually. Some kind of garbage collection must be provided.

Another interesting exception is that a look-ahead control agent might arrive a node later than the target agent. (This can be detected from the execution log.) In that case, a chase algorithm must be activated to chase the target agent along its traveling trace.

## 7. Concluding Remarks and Future Work

A ubiquitous information service environment needs to offer the client mobility allowing its client to move across different logical networks. To overcome the intermittent connection problem inherent in mobile environments, it also needs to offer the agent mobility allowing its clients to access network services by sending an intelligent message to the network. To guarantee the service quality for a service network, an OA&M system is needed to manage both mobilities. This paper proposes a hybrid OA&M infrastructure that can incorporate both centralized and distributed supporting facilities to offer an effective and economic OA&M support to both mobilities. This infrastructure makes use of personal Internet facilities to share the expensive OA&M cost. Based on the proposed infrastructure, several solutions are proposed to solve such problems as agent search and agent control. **All proposed solutions require no change to the current communication networks, and are very easy to implement.**

A quantitative analysis will be needed to determine which of those proposed solutions are more appropriate in various situations. Nevertheless, there are many other problems such as transaction control, resource control, security, etc. yet to be addressed to make a ubiquitous information service commercially viable.

## References

1. T. Alanko, et al., "Measured Performance of Data Transmission Over Cellular Telephone Networks," *ACM SIGCOMM Computer Communication Review*, pages 24-44, August 1994.
2. J. F. Bartlett, "W4 - The Wireless World Wide Web," *Proc. of the IEEE Workshop on Mobile Computing Systems and Applications*, December 1994.
3. T. Berners-Lee, et al., "The World-Wide Web," *Communications of the ACM*, vol. 37, no. 8, August 1994, pp. 76-82.
4. T. Berners-Lee, R. Cailliau, J.-F. Groff, and B. Pollermann, "World-Wide Web: The Information Universe," *Electronic Networking: Research, Applications and Policy*, vol. 1, no. 2, Meckler, Westport, CT., Spring 1992.
5. G. H. Forman and J. Zahorjan, "The Challenges of Mobile Computing," *IEEE Computer*, March 1994, pp. 38-47.
6. J. Garrahan, P. Russo, K. Kitami, and R. Kung, "Intelligent Network Overview", *IEEE Communications*, March 1993, pp. 30-36.
7. Michael Genesereth and Steven Ketchpel, "Software Agents", *CACM*, July 1994, pp. 48-53.
8. S. Gessler and A. Kotulla, "PDAs as Mobile WWW Browsers," *The Second World Wide Web Conference '94*, October, 1994.
9. T. Imielinski and B. R. Badrinath, "Mobile Wireless Computing: Challenges in Data Management," *Communication of ACM*, August 1994.
10. M. F. Kaashoek, T. Pinckney, and J. A. Tauber. "Dynamic Documents: Mobile Wireless Access to the WWW," *Proc. of the IEEE Workshop on Mobile Computing Systems and Applications*, December 1994.
11. K. D. Kotay and D. Kotz, "Transportable Agents," *Proc. of the Third International Conference on Information and Knowledge Management*, December 1994.
12. James Lee, "Intelligent Messaging Paradigm and Telescript," *CCL Technical Journal*, No. 35, Dec 1, 1994, pp. 37-41.
13. Ichiro Lida, Takashi Nishigaya, Koso Murakami, "DUET: An Agent-Based Personal Communications Network", *IEEE Communications*, Nov. 1995, pp. 44-49.
14. Yao-Nan Lien, "Perspective of Service Networks on National Information Infrastructure," *CCL Technical Journal*, No. 35, Dec 1, 1994, pp. 28-36.
15. Yao-Nan Lien, "An Open Intelligent Messaging Network Infrastructure for Ubiquitous Information Service," *Proc. of the First Workshop on Mobile Computing*, Hsing-Chu, Taiwan, April 1995, pp. 2-9.
16. Yi-Bin Lin, "Determining the User Locations for Personal Communications Services Networks", *IEEE Transactions on Vehicular Technology*, vol. 43, no. 3, Aug. 1994, pp. 466-473.

17. P. Maes, "Agents that reduce work and information overload", *CACM*, July 1994, pp. 30-41.
18. Jay Padgett, Christoph Gunther, Takeshi Hattori, "Overview of Wireless Personal Communications ", *IEEE Communications*, Jan. 1995, pp. 28-41.
19. Charles Perkins, Pravin Bhagwat, "A Mobile Networking System based on Internet Protocol", *IEEE Personal Communications*, vol. 1, no. 1, 1st Qtr. 1994, pp. 32-41.
20. Charles Perkins, Kevin Luo, "Using DHCP with computers that move", *ACM Wireless Networks*, vol. 1, 1995, pp. 341-353.
21. R. J. Vetter, C. Spell, and C. Ward, "Mosaic and the World-Wide Web," *IEEE Computer*, October 1994, pp. 49-57.
22. G. M. Voelker and B. N. Bershad, "Mobisaic: An Information System for a Mobile Wireless Computing Environment," *Technical Report, Department of Computer Science and Engineering, University of Washington*, September 1994.
23. T. Watson, "Wit: An Infrastructure for Wireless Palmtop Computing," *Technical Report UW-CSE-94-11-08. University of Washington*, November 1994.
24. M. Weiser, "The computer for the 21st century", *Scientific America*, 1992, pp. 94-104.
25. James E. White, "Telescript Technology: The Foundation for the Electronic Marketplace", General Magic, Inc.
26. Mohammed Zaid, "Personal Mobility in PCS", *IEEE Personal Communications*, vol. 1, no. 4, 4th Qtr. 1994, pp. 12-16.
27. TIA/EIA IS-41, "Cellular Radio Telecommunications Intersystem Operations", *Telecommunications Industry Association*, Dec. 1991.