Applying Human-like Intelligence to Future Generation Network to Improve Communication Efficiency

Yang Li

This thesis is submitted in partial fulfillment of the academic requirements

for the degree of

Doctor of Philosophy in Electrical Engineering

in the Faculty of Engineering and The Built Environment

University of Cape Town

December 2007

As the candidate's supervisor, I have approved this dissertation for submission.

Name: Professor H. Anthony Chan

Signed: ______________________

Date: ________________________

Declaration

I hereby declare that: (1) the above thesis is my own unaided work, both in conception and execution, and that apart from the normal guidance of my supervisor, I have received no assistance apart from that stated below; (2) except as stated below, neither the substance or any part of the thesis has been submitted in the past, or is being, or is to be submitted for a degree in the University or any other University.

I am now presenting the thesis for examination for the Degree of PhD. I also grant the University free license to reproduce the above thesis in whole or in part, for the purpose of research.

Yang Li

Name

Date

To my darling husband – Tao Sun

Abstract

In recent decades, communications network has evolved at drastic speed to provide advanced and intelligent services. This strengthening service provision owes to the successful establishment of various intelligent networks and the use of artificial intelligence, pervasive computing, and social networking in communications. It has consequently endowed network users with abundant choices of communication services.

While these communications services are bringing convenience to human lives, people in turn are performing more tasks. The current network with its large number of available communications services is then often burdening network users with the complexity and inflexibility in using these services. In particular, the network lacks the initiative and the ability to investigate a user's most recent communication needs and subsequently adjust the manner of service provision according to these needs and user connecting possibilities. The network needs to be more intelligent to handle these problems.

We therefore propose importing human-like intelligence into the network to facilitate communication-session processing according to user needs. Enhanced with the intelligence, the network should then be able to provide intelligent solutions to connectivity choices in the case of multiple possibilities connecting users. Hereof, human-like intelligence is the user's ability to identify an occurring communications problem, analyze the properties of the problem, and propose solutions to solve the problem. If the network is able to investigate communications problems and propose smart solutions on identifying optimal service-delivery manner, it is factually more intelligent and should be able to carry out communications according to user needs, thereby relieving users from heavy communications burden.

The key questions for the network to possess intelligence are how to investigate the occurring problems and what rules to use for suggesting optimal solutions. To solve these problems, we establish a virtual-user system to work on behalf of human-like intelligence in the network. A virtual user uniquely represents a real user by physically storing his/her communication profile (personal details, social relationships, and communication schedules) in the network. The system then determines the optimal session-delivery manners according to this user information and attempts to best meet the user needs. The potential manners comprise the general ones (i.e., the manners of *fail* and *deliver*), those obtained through extensive network resources (i.e., the manners of *postpone* and *force*), and those obtained through recruited human resources (i.e., the manners of *help* and *learn*).

The system has been successfully implemented in a Java/MySQL software environment. It is then successfully tested for module functionality, intelligence decision-making ability, and social-knowledge utility. It is also able to improve communication efficiency by increasing session-delivery success rate, balancing network traffic, optimizing network-resource utilization, and successfully conducting several user abilities. The system is therefore quite practical.

In conclusion, we envision entertaining the users with a considerate and smart communications network by enhancing it with human-like intelligence. The virtual-user-system approach then physically implements this vision by providing the following significant functions. First, it improves the communication efficiency by making use of users' social features such as two users' mutual trust degrees and each user's trustworthiness. Second, it reduces real users' communications workload through the interaction of virtual users and the network. Third, it creates an environment to merge user human natures into user communications by representing these natures as network characteristics and quantifying the properties of the natures as network-recognizable characteristic values.

Acknowledgement

First and foremost, I am deeply indebted to my supervisor Professor H. Anthony Chan for his genuine concern, encouragement, support, and advice in my research. His specific "sky view" approach applies to my technical writing as well as personal development as an academic.

The laboratory of the Communications Research Group provided an ideal research environment for my graduate study at University of Cape Town (UCT). I am particularly grateful to Mr. Neco Ventura for his steady hand in directing the laboratory and ensuring its practice.

Our department possesses a praiseworthy atmosphere of academic excellence combined with friendliness and openness. My gratitude goes to Professor Mqhele Dlodlo and Professor Trevor Gaunt for intellectual stewardship, Dr. Andrew Wilkinson as an example of admirable researcher, and Mr. Mputumi Nkewu for generous help. Thanks to Mrs. Nicole Moodley for enjoyable chats, Mrs. Cosmos Shatto for pleasant cooperation, Mrs. Marlene Joubert and Mrs. Lesley Rossouw for assistance in financial issues, and Mrs. Karin van Wyk for friendliness. I am also in debt to many other departments at UCT for their support over the last two and half years.

I have also benefited from the time spent with my UCT colleagues. Special thanks to Deeya Nursimloo for her sweetness, Benon Muwonge for his understanding, Dali Wei for conversations in our mother tongue, George Kalebaila for in-depth chats, Indika Piyasenna for cool music, Zoran Basich for fantastic parties, Pascal Acquah for birthday wish, Kamil Hussien for interesting discussions, Vitalis Ozianyi for unselfish help, Samuel Senkindu for trust, and Oladayo Salami for encouragement. Their dedication to research kept me disciplined and their encouragement lightened my life. Extra thanks to all the undergraduates with whom I worked.

I truly appreciate the effort that Mr. Mike Crooke, Mrs. Lorna Solorman, Mrs. Sally Burt, and Mrs. Jane English have made in editing my thesis and technical papers.

I am particularly fortunate to have Bill Tucker and Lizanne Tucker as good friends, who never hesitate to give a helping hand in my life; especially to Bill as a frequent counterpart for Wednesday lunch conversations. Special thanks go to my Masters supervisor Professor Johnson Agbinya (now a lifetime friend) for his continuous interest in my research. I am also grateful to Tracy October for sharing life experiences and Dave Liu for caring of my thesis progress.

My husband, Tao Sun, deserves my love in our entire marriage life for his unselfishness, positive encouragement, and taking care of me. Special thanks to our parents who have brought us into this wonderful world. My father is the forever supporter of "all" my decisions and my mother always attempts to lure me home with delicious hometown food. Great thanks to my mother-in-law, father-in-law, and brother-in-law for their inspiration and enlightenment.

This research was supported in part by Broadband Network Center of Excellence at UCT with funding support by Telkom South Africa, Nokia-Siemens Networks, Telesciences, and Thrip.

Table of Contents

List of Figures

List of Tables

Glossary

ANSI-41 (*section 2.5.1.2***)** is a network standard that identifies and authenticates users, and route calls on mobile phone networks by allowing information exchange on switches [66].

Application (*sections 3.2.2* and *4.1.2*) is a physical realization of a network service on specific devices and links.

Communication Event (*section 1.1*) is a scenario where a user establishes a connection with another via the communications network to fulfill a specific task. The user who sets up the connection is called "initiator" and the one being reached through the connection is called "receiver".

Communicating Manner (*section 1.1*) is a combination of four communication dimensions that include time, communicating users, user locations, and user devices.

Communication Resources (*section 3.2.2*) comprise network resources, human resources, and information resources in the intelligence network.

Communication Service (*sections 1.1, 4.1.2***)**, frequently short as "service", comprises the task that a communication session accomplishes and the function that the session provides to users.

Communication Session (*section 1.1***)**, or "communication event" expressed in a more technical way, is a lasting connection between two users, involving the exchange of information between their communication devices. The connection starts from an initiator and ends at a receiver.

Connecting Manner (*section 4.1.2*) is a set of physical devices and links for session execution.

Customized Applications for Mobile network Enhanced Logic (CAMEL, *section 2.5.1.2*, [62][63]) is the technical realization of a 3GPP initiative to extend traditional IN services found in fixed networks into mobile networks by adding in roaming services.

Database Management System (DBMS, *section 5.1.2***)** is a collection of software for organizing the information in a database, concerning data input, verification, storage, and retrieval [89].

Dynamic Service-Session Database (DSS-DB, *section 4.2.5***)** is a network database used to store the real-time changing communication information of users.

Enhanced Data rates for GSM Evolution (EDGE) is a digital mobile phone technology that allows increased data transmission rates and improved data transmission reliability in the same spectrum allocated to 2G systems to align the technology with 3G systems.

Event Involver (section 1.3.4), also called "event participator", is a network user who participates in a communication event as the initiator, the receiver, or the assistant communicating party.

Expert System (*section 2.3.1*) is a computer program that uses what appears to be reasoning abilities to reach conclusions according to the stored knowledge of specific subjects and that of human experts [35].

Future Generation Network (FGN, *section 3.2***),** short for "Human-like Intelligence enhanced Future Generation Network", is an instance of the intelligence network.

General Network (*section 3.2.2*) is the current communications network without human-like intelligence embedded. It is often referred to when compared with an intelligence network.

General Packet Radio Service (GPRS, *section 2.5.1.2***)** is a mobile data service available to GSM users and IS-136 users [61].

Graphic User Interfaces (GUI, *section 6.4.1***)** is the term for graphic user interfaces to a computer. A user can control computer programs by using the icons, buttons, and pointers on the interfaces [101].

Human-like Intelligence (HI, *section 1.3***)** is, in semantics, an ability to solve problems, to identify new problems and analyze them, and to contribute valuable inventions and services to their solution or, in techniques, the ability to recognize, apprehend, plan, troubleshoot, abstract-ideation and thought, express, speak in several languages, and learn.

Human-like Intelligence-enhanced Future Generation Network (HIFGN, *section 3.2***)** is the next generation network that uses Human-like Intelligence to smartly plan and execute communication sessions according to users' personalized requirements and to independently perform communication sessions with no need for redundant personal efforts from users.

Human Resources (*section 3.2.2*) include network users, their communication devices, and their participation in communications.

Human-intelligence Part (*section 3.2.2*) is an area of hardware that stores the mechanisms with which the network implements human-like intelligence in the form of software. It is often referred to when compared with the physical-network part in the intelligence network.

Institute of Electrical and Electronics Engineers (IEEE) is a professional organization whose activities include the development of electronics, communications, and network standards.

IEEE 802.11 is an IEEE technical standard covering Wireless-Local-Area-Network technology and is mainly categorized into two groups of standards – 802.11a and 802.11b. IEEE 802.11 networks are composed of stations, wireless medium, access points, and a distribution system.

IEEE 802.11a is part of the IEEE 802.11 family of specifications. It operates in the 5GHz unlicensed band and supports data rates up to 54Mbps.

IEEE 802.11b is part of the IEEE 802.11 family of specifications. It operates in the 2.4GHz unlicensed band and uses High-Rate-Direct-Sequence-Spread-Spectrum technology to achieve a data rate of up to 11Mbps.

Information Resources (*section 3.2.2*) are the information contents carried by communication sessions. They come into being with the birth of these sessions and vanish when the sessions are over.

Integrated Development Environment (IDE, *section 5.1.3***)** is a programming environment packaged as an application program. It typically consists of a code editor, a compiler, a debugger, and a GUI builder [96].

Intelligence Network (*sections 2.5.1.1* and *3.2.2*) is a future generation network that uses human-like intelligence to help with decision-making for the execution of communication sessions. It is often referred to when compared with a general network.

Intelligent Network (IN, *section 2.5.1.1***)** is a network architecture that allows network operators to provide value-added services in addition to the standard telecommunication services [54] .

Intelligent Network Capability Set (INCS, *section 2.5.1.1*) is the standard that regulates the capabilities that IN can provide [54]-[58].

International Mobile Telephony 2000 (IMT-2000) is a vision for a single global standard for heterogeneous wireless networks and it is proposed by the ITU-T.

International Telecommunication Union-Telecommunication (ITU-T, *section 3.1.1***)** is an international organization that develops worldwide standards for information and communications technologies [53].

International Telegraph and Telephone Consultative Committee (CCITT, *section 2.5.1.1***)** is the primary international organization for fostering cooperative standards for telecommunications equipment and systems [53].

Internet Protocol (IP, *section 1.2.1***)** is a data-oriented protocol used for communicating data across packet-switched computer networks [49] .

IP Multimedia Subsystem (IMS, *section 2.5.1.3***)** is an architecture that enables support for IP multimedia applications within the UMTS system [70][71].

Interim Standard 136 (IS-136) is often referred to as Digital Advanced Mobile Phone Service, with the system using a Time-Division-Multiple-Access process over the radio interface.

Java (*section 5.1.1*) is a platform-independent object-oriented programming language developed by Sun Microsystems and used for writing interactive codes [86].

Java Database Connectivity (JDBC, *section 5.1.3*) is a Sun-Microsystems interface standard that defines how Java applications access data from relational databases [91].

Matrix (*section 7.3*) is an array of elements and is generally used for linear transformation in mathematics and statistics. The type of these elements can be string, number, or any quantity [102].

Middleware (*section 2.5.1.3*) is software that mediates between two disparate application programs across diverse computing platforms and networks [70].

MySQL (*section 5.1.1*) is an open-source programming language developed for a relational database management system that adds, accesses, and processes data in a database [90].

MySQL Connector/J (*section 5.1.3*) is a JDBC-3.0 type-4 driver that uses pure Java, implements the version-3.0 JDBC specifications. It directly communicates with the MySQL server using the MySQL protocol [92].

Network Involvers (*Chapter 1*) refer to different groups of people who relate to the communications and computer network in certain ways, including users, operators, service providers, and researchers.

Network Resources (*section 3.2.2*) include the equipment, technologies, and standards for a computer/communications network.

Network Service (*section 3.2.2*) is a process that creates benefits by facilitating a positive change in network users' communications. In a narrow sense, a service means a function that the network offers to meet the communication requirements of a user.

Network Traffic (*section 7.3*) is a series numbers of communication sessions that a network requests or actually delivers at evenly-distributed successive moments.

Network User (*section 3.2.2*) is any social entity in reality, including a private person, a group of people, a terminal, an IP address, a technology, a network, or software. In the thesis, it is often referred to as a real user.

Physical Location (*section 5.3.1*) is a geographic position where a communication event occurs. It relates closely to a user's communication schedule at that moment.

Physical-network Part (*section 3.2.2*) is the term for the current communications network. It is often referred to when compared with the human-intelligence part in the intelligence network.

Poisson Distribution (*section 7.2.1*) is a discrete probability distribution that expresses the probability of a number of events occurring in a fixed period if the event arrival rate is fixed and the event occurrence is time independent [102].

Principal User (*section 4.1.1*) is the core user whom we need to analyze when establishing social relationship information in the intelligence network.

Public Land Mobile Network (PLMN, *section 2.5.1.2*, [62]**)** is a generic name for all mobile wireless networks that use land-based radio transmitters or base stations.

Quality of Service (QoS, *section 3.1.1***)** guarantees a certain level of a specified resource to selected traffic on a network, such as a maintained level of bandwidth or low latency [75] [76].

Second Generation (2G) is the second generation of mobile communication systems that mainly provide voice services using digital circuit-switched technologies.

Service Session (*section 3.2.2*), or "communication session", is a lasting connection between a user that requires a communication service and a provider that offers the service. It is an instance of the communications network delivering a service to network users.

Session Arrival Rate (*section 7.1*) is the number of session-service requests per unit time. In the thesis, it is defined as the request number generated at the starting moment of the unit time.

Session Delivery Rate (*section 7.1*) is the number of actually delivered sessions per unit time, where these sessions have initially been requested at the starting moment of that unit time.

Session Execution Time is the time taken by the physical network to actually execute a communication session.

Session Involver, also called "session participator", is a network user who participates in a communication session as the initiator, the receiver, or the assistant user.

Session Priority (PRI, *sections 4.2.5 and 4.5.2***)** is a term specifically defined in the Session-Keeper module of the virtual-user system. It determines how fast a service session can reach the point of being immediately processed.

Session-delivery Success Rate (*section 7.2*) is the number of successfully delivered sessions as a percentage of the number of expected-to-be-delivered sessions in the same period.

Session-processing Stage (*section 6.1*) is an in-process stage where the virtual-user system has just registered the session and performs the session-generating, comparing, and decision-making for the first time.

Session-reprocessing Stage (*section 6.1*) is an in-process stage where the virtual-user system has determined to postpone the session and repeats the process of generating, comparing, and decision-making only when necessary.

Session-executing Stage (*section 6.1*) is an in-process stage where the virtual-user system has successfully determined the final session-delivery manner.

Social Domain (*section 5.3.1*) is a metaphorical field where a group of social individuals share characteristics or knowledge, have common interests, performs similar communication tasks, and influence each other's communication status in a certain way.

Social Relation (*section 1.2.1*), also expressed as "social relationship", refers to a multitude of interactions between two social members, with each taking up a social position and conducting a set of specific social behaviors.

Structured Query Language (SQL) is syntax for defining and manipulating data from a relational database developed by IBM.

The Global System for Mobile communications (GSM, *section 2.5.1.2***)** is a second-generation cellular telecommunication system that transmits digital information in both signaling and speech channels. It therefore has introduced many enhancements such as security, capacity, quality, and the ability to support integrated services [60].

The principal user (*section 4.1.1***)** refers to the core user whose social relationships are under investigation in a specific communication scenario.

The range (*section 4.5.1***),** sometimes expressed as "[0, 100]", is a natural number range between zero and a hundred.

Third Generation (3G) is the term given to the third generation of mobile communication systems that offer enhanced multimedia services (especially the merged voice and data) through improved spectrum efficiency.

Third Generation Partnership Project (3GPP, *section 1.2.1***)** is the collaboration between groups of telecommunications associations that produce globally applicable Technical Specifications and Technical Reports for 3G based on the evolved core networks of GSM and the radio access technologies of UTRA, GPRS, and EDGE [104] .

Third Generation Partnership Project 2 (3GPP2, *section 1.2.1***)** is the collaboration between telecommunications associations to make a globally applicable 3G mobile phone system specification within the scope of the ITU's "IMT-2000" project [105].

Telecommunications Industry Association (TIA, *section 2.5.1.2***)** is a trade association that delegates global ICT industries in many ways and especially enhances the business environment for telecommunications companies [67].

Universal Mobile Telecommunications System (UMTS) is a type of 3G mobile-communications system that provides an enhanced range of multimedia services, with its specifications formulated by 3GPP.

Universal Terrestrial Radio Access (UTRA) a technology that identifies the Frequency-Division-Duplex mode and Time-Division-Duplex mode of access for the UMTS system.

User Intelligence (*section 3.2.2*) is another way of presenting "human-like intelligence".

Virtual Session (*Chapter 4 and section 4.1.3*) is the image of a real communication session in the virtual-user system through a course of software programs. It possesses the major features and essential functions of the real session.

Virtual User (*section 3.3.1*) is the representative of a real user in the intelligence network. It maintains the real user's communication profile in terms of personal details, current communication status, and social relations.

Wireless Fidelity (WiFi) is an interoperability standard developed by Wireless Ethernet Compatibility Alliance and issued to those manufacturers whose IEEE 802.11a and IEEE 802.11b equipment has passed a suite of basic interoperability tests.

Chapter 1 Introducing Human-like Intelligence to Network

With the globalization of information exchange in recent decades, people are contacting each other more often and in more ways, regardless of their individual social roles, acquaintance, and mutual attitudes. These objective social factors, however, add to the burden of communication between people to certain extent. People need to fulfill the requirement on each of these factors that contribute to complete a communication event (section 1.1). Unfortunately, most current communication designs have not fully considered human sociality, as they generally only address technical problem solutions (section 1.2). It is thus desirable to accommodate more human-related factors, such as the mutual trust fidelity between two communicating parties, into communication system design. Meticulous proposals and corresponding approaches are necessary to technically support such a design (section 1.3). In theory, this design will facilitate the humanization of modern communications so that the resulting network may deal with communicating users in a more friendly manner (section 1.3.4).

1.1 Heavy Burden on Users by Modern Communications

The rapid development of the communication industry has created a communicating world where one can easily reach others and be easily reached by them. This efficient communication environment speeds up the evolution of society as a whole and empowers the development of other industries such as commercial activities. Therefore it has directly and indirectly created plenty of job opportunities for people and improved their quality of life. However, people in such an environment are also getting busier and busier with endless phone calls, short messages, electronic mails (e-mail[1] [1]), and Internet[2] ([2]) online advertisements.

[1] E-mail is an application that uses store-and-forward method to compose, send, store, and receive messages over electronic communication systems.

This large number of communication events[3] has caused certain burden and inconvenience in users' communication aspect of life.

Firstly, with regard to some events that do not necessarily require a user's participation, the user may get an invitation to the event from unknown sources and get involved before realizing that he/she does not need to do so. For example, Lisa gets an incoming call with unknown caller's number for selling a product that Lisa does not need (getting an invitation to an unnecessary event). Without knowing that the call is an advertisement call, Lisa may answer the call (getting involved before realizing the truth). Lisa's life is in fact disturbed because she is unaware of the communication contents.

Secondly, users are suffering the inconveniences caused by real-world communication services, instead of enjoying services accessible "anywhere, anytime, by anyone" and in any form [3]. (1) Current network merely distributes services to fixed devices at a specific location according to users' preset profile. Unless users have subscribed to call-forwarding in advance, the network will not re-direct the call according to the time and locations supplied by the user. In addition, the network does not take the initiative to analyze user mobility and take initiative to direct the service to the most convenient device for users. For example, company salesperson Lisa is out of office for an exhibition. Current communications network will not automatically switch the phone calls going to Lisa's office phone to her cell phone because it cannot find out that Lisa is unavailable for her office phone without Lisa having to manually input such information in advance. (2) Users may not be able to receive services when they are personally busy with other social events, even though they are physically ready for the incoming services. For example, when Lisa is in office, she cannot answer the call to her office phone if she is answering another call on her cell phone. (3) Users initiate a communication event in order to

[2] Internet is a worldwide, publicly accessible computer network that transmits data by packet switching using the standard Internet Protocol and allows Internet Service Providers to manipulate the data.

[3] Communication event is a scenario where an initiator establishes a connection with a receiver via the communications network to fulfill a specific task.

implement a social task. Only the users designated in the event are able to receive the communication service and consequently to execute the task. Other users who get no permission of processing the communication event cannot take over the task even though they are personally able to accomplish the task. For example, the call to Lisa's office phone is a general enquiry. Her colleague, Bill, is therefore able to answer the call by authority. However, because Lisa does not forward the call to Bill and Bill does not know that the call comes from a general enquiry phone, he cannot answer the call for Lisa. (4) Users have limitations in choosing their favorite type of communication manners[4] at a given time. The current communications network is not able to provide such multiple choices for users because it cannot reflect users' real-time communication information. Though two types of device belong to the same user, the network cannot automatically switch services that are supposed to reach one type of device to another type of device. For example, when delivering a service to Lisa, the system is not intelligent enough to choose the available device between Lisa's office phone and cell phone. This is because the network cannot detect two facts: whether both phones belong to Lisa and which of the two phones is available at that time. Even if the network can do the above two things, it has no right to act against the original requirement from the call initiator and assertively switch the call to Lisa's available device.

Thirdly, users also find it confusing to use new generation devices that claim to be able to provide a variety of advanced services. Most users are not specialists in engineering or science and they are not familiar with new technologies and frequent updates. Even after having carefully read the manuals, users still find it annoying to go through those complex set-up steps in order to activate the services. This problem exists especially amongst the elderly because they can hardly follow the fast voice prompt, they cannot easily find the proper keys, and some of them even refuse to use new technologies. Even if the young generation is capable of manipulating complex services, they do not have to take effort to do so if the network could step in.

[4] Communicating manner is a combination of four communication dimensions that includes time, communicating users, user locations, and user devices.

Fourthly, service users have no access to the core of current communication services[5] but have to use the existing ones developed by network players. Users may personalize their profiles (i.e., choosing their favourite interfaces for operating services) and may select their favorite set of services (i.e., selecting Vonage features [4]). Yet only network operators and service providers are primarily involved in the rules, design, operation, and management of these services. In addition, the use of communication services is at best transferable with restriction. That is, if a user is not using a type of service, his/her friend cannot share with him/her the interesting contents that are presented through the service. Notice that this argument is not against the generally accepted concept of "network value", which claims that the value of the telecommunications network increases with the number of users [5]. The network-value theory builds upon the fact that users are all using a service, and the benefit from using the service is then expandable and transferable among the users. Yet in the previous case, the user's friend attempts to share contents with the user through some service that the user is not using. It would then be unacceptable if the network forces the delivery of the service to the user without his/her permission. With this worry in mind, the transfer of service is considered restricted.

Last but not least, there is no intelligent guide for users from the network side on how to perform a communication event most efficiently. The event initiator has to try all the contact manners of the receiver to get hold of him/her because the initiator has no idea what the most appropriate device is for the receiver at that moment. Yet it is easier for the network to detect and to find out the easiest communication manner from the initiator to the receiver because the network is able to real-time track the receiver's status.

The above analyses present to us that, apart from providing users with various novel communicating manners, modern communications are still bothering users' life in the following respects. (1) Currently, communications have overloaded users' life with volumes of meaningless information because the network cannot tell the difference between the relevant and irrelevant

[5] Communication service, frequently short as "service", comprises the task that a communication session accomplishes and the function that the session provides to users.

communication contents for the users. (2) With the provision of a limited number of communicating manners, the network has factually caused inconveniences to users' social life by constraining them from choosing their preferred communication manners. (3) Modern network users are hassled more in their daily life, because they are equipped with many communication devices and are thus available in more ways in theory. (4) Users do not have enough autonomy and freedom to operate on and manage the process of services that are applicable to their needs. (5) Users cannot react to the changing communication environment as fast as possible during a communication session due to the lack of proper guidance from the current network.

1.2 Problems of Existing Network without Sufficient Human Factor Consideration

In enjoying the novelty and variety of modern communications, users also suffer from the overload, inconvenience, inflexibility, autarchy, and remoteness of the communications. A great number of technical reasons are responsible for users' miseries. This project uncovers the problems that lack sufficient consideration about network users' preferences, comforts, and desires in the design and realization of the communications network.

1.2.1 Problem Statement: Desired Network Improvement with Human Perspectives

As described in section 1.1, modern communications with more and more new and advanced services can burden users. Possible improvement is for the communications network to exhibit human perspective. It is desired that the network is able to accommodate the needs of users in respect of what they like and dislike, what they are able and unable to do, what they do and do not prefer to do. Lacking careful consideration about user preferences can result in many unsolved technical problems in network design. The advantages for the network to process human perspectives are briefed below (see appendix A.1 for detailed explanation and examples).

(1) Practical user-centric service platform that is deliberately developed for seamless use under different network vendors [6] has plenty of room for improvement.

(2) Current communications network will be able to detect several man-made uncertainties and therefore offer proper solutions to reduce the damage to communication events caused by these uncertainties.

(3) Current communications network will be able to deliver better service to users if taking the initiatives to identify potential connecting manners for users.

(4) Current communications will be able to make more efficient use of each of the network-, human-, and information-resources[6] and create a mechanism to interact between them.

(5) Current network can avoid sacrificing users' first desire to solve the technical problems encountered

(6) The wasted resources of communication society, in which thousands of users are repeating the same operation steps in order to use the same service, can be reduced.

(7) In failure analysis and corrective actions for unsuccessful communication events, most studies have been focusing on the technical problems arising from the physical network such as congested traffic and information protection. It is desired to extend these studies to include the unavailability of desired communicating parties (e.g., expected session receiver) or the underdeveloped potential distribution of human resources over the network.

(8) Customer satisfaction can be increased. Network operators may lose current customers if the operators cannot meet users' individual requirements on service. User-centric service-provision platforms enables the network to attract and enroll new customers in exploring new types of service under different access networks.

1.2.2 Motivation: Optimizing Network to Relieve Users of Heavy Communication Tasks

[6] Refer to section 3.2.2 for the definition of network-, human-, and information-resources.

In order to enjoy the diverse multimode services, it is desirable that the users do not need to endure the overload, inconvenience, inflexibility, and restriction of the large number of services offered by the communications network itself. Lack sufficient consideration about the users' communication needs in network design causes users' dissatisfaction. It is therefore necessary to optimize the existing network to relieve users from the communication burden and answer their personal needs to the best.

1.3 Proposals to Apply Human-like Intelligence to Network

To fit in users' shoes, the network is supposed to be able to observe users' actions during events, to infer their future needs on new events, to understand their ways of thinking, to anticipate their decisions, and to create new methods for processing events in a more user-friendly manner. Once equipped with these powerful abilities, the network is then able to provide users with a service equivalent to that provided by a human operator. The enhanced network knows what users like and dislike, how users generally behave during communication, how they react to an instantly changing communication environment, and how they interact with each other.

In this regard, we hypothesize (1) that the human-intellectual-ability enriched network is able to assist users with their communications in a smart and humanized manner and (2) that it is able to ensure users a joyful communication life with fewer discomforts than those users are presently tolerating. We propose applying human-like intelligence to the communications network so that the human-like intelligence embedded network is able to serve users like a considerate human.

Different from most sources that define human-like intelligence as a collection of intelligence acquired by or derived from human resources, we define human-like intelligence[7,8]

[7] "To my mind, a human intellectual competence must entail a set of skills of problem solving - enabling the individual to resolve genuine problems or difficulties that he or she encounters and, when appropriate, to create an effective product - and must also entail the potential for finding or creating problems - and thereby laying the groundwork for the acquisition of new knowledge." – Howard Gardner.

([7] [8]) in semantics as an ability to solve problems, to identify new problems and analyze them, and to contribute valuable inventions and services to their solution. For easier analysis and emulation, we express human-like intelligence as eight types of technical ability: the abilities of (1) recognition, (2) apprehension, (3) planning, (4) troubleshooting, (5) abstract ideation, (6) expression, (7) languages, and (8) learning, and call these abilities "human-like intelligence". Once incorporated with these abilities via suitable mechanisms, the network is expected to solve communication problems similar to that of a human.

Besides the vision of applying human-like intelligence to the network, the following questions need to be answered to implement the application. Is it technically possible to import a human being's social abilities into a mechanical network? If possible, what human factors account for much in network humanization and how many can we use for the network? How do we expect the network to behave when it works with selected human factors? What appropriate methodologies can we use to testify to the prospective behaviors of the network and to illustrate the simulation of such tests in lab/experiment/real environments? Is the human nature planted network capable of handling complex communication events alone? Can the network benefit users as expected? The following sections will answer these questions in sequence.

[8] "... the aggregate or global capacity of the individual to act purposefully, to think rationally, and to deal effectively with his environment." – David Wechsler.

1.3.1 Approach: Mapping Human Intellectual Behaviors and Social Relations onto Network

To introduce human-like intelligence into the communications network, differentiating the communication between people from that within the network stands in breach. First, people and a network complete a communication session for different purposes. People communicate to fulfill an explicit task such as negotiating the price for a product, whereas the network performs a communication event to accomplish the task assigned by the event initiator (e.g., a communication-service user). Second, because of the different communication purposes, people and the network place different emphasis on a communication session. People normally raise a problem at the beginning of the session and solve the problem in the end. The network simply goes through the session by following what the initiator has asked for. Third, people's communication may involve several participators and these participants can join in and leave anytime during the session, whereas the number of communication participators in the network is normally fixed in the process of communication. To sum up, humans behave cognitively during communication whereas the network operates rigidly.

Two factors are conclusively responsible for the cognition of human communication: human intellectual behaviors and human social relations[9] ([9]). Using intellectual behaviors, humans are quite capable of solving problems. They detect a problem, raise the problem as an interesting topic, discuss the topic, and draw conclusions. Through social relations, humans are able to bring new human resources into their communication sessions. They may refer to knowledgeable people who are not involved in the session for a broader range of information. This behavior would subsequently increase the possibility of getting the session successfully done.

---

[9] Social relation, also called "social relationship", generally refers to a multitude of interactions between two members, with each taking up a position and conducting a set of specific behaviours in a human society.

We therefore endow the network with intelligent abilities by mapping network users' intellectual behaviors and social relations onto it. We establish a virtual user in the network to work side by side with the existing network components. This virtual user possesses a real user's intelligence and works on behalf of the real person when dealing with a communication session. It keeps the real user's communication profile, detects problems when the required facts of a communication procedure deviate from the real user's original desires, makes decisions according to the real user's preferences, and relates to other virtual people via the social relations of the real user. These mappings enable the virtual user to behave intelligently like a real person in all communication sessions.

With respect to function description, these virtual users act on behalf of real network users in processing communications events; with respect to physical realization, these virtual users are electrical devices (i.e., computers) that exchange specific data for implementing human-like intelligence. The more complex communicating situation these intelligent devices can handle, the more they can relieve the real users from multifarious impersonal communications.

1.3.2 Objectives: Enhancing Human-Network Interaction

Of course, simply inserting human-like intelligence into the network will not make the network work any better for users, unless the inserted intelligent network components collaborate with the existing ones. (1) Without the intelligent components' prompt, the normal components can barely reflect users' intelligence. They have no idea what a user requires at a given moment, when the user schedules a change to his/her communication status, or whether the user would like to involve more communicating partners. (2) Similarly, the intelligent network components are only responsible for analyzing the users' communication status, making proper decisions, and real-time instructing the existing network parts to execute users' demands. They have nothing to do with the network's physical connections and disconnections.

Thus, the objectives here are to enhance the interaction between the users and the network and to make the best out of their cooperation via the mutual network component – the virtual user. Virtual users make it possible (1) for the network to inform the users of network changes and of available network resources and (2) for the users to present their preference to the network during service execution. This research addresses the proof-of-concept part of the approach.

1.3.3 Methodologies: Design and Realization of Human-like Intelligence Embedded Network

To determine whether the hypothesis of applying intelligence to the network works, we have come up with the approach of setting up virtual people to work on behalf of real people during communication. This method is called *concept interpretation*.

The approach is then simulated in a software environment. If the simulation results show that the selected approach is able to enhance network performance and the amount of enhancement overcomes the extra efforts that the network has to endure, it can be deduced that the hypothesis is correct when using the virtual-user approach. Whereas if the simulation results show no improvement in network performance or the cost on realizing the approach is much higher than the amount of benefit that the network gets, the hypothesis then proves to be impractical when working with the selected approach. No matter whether the results are positive or negative, this method is called *simulation approval*.

Testbed experiments are a way to testify to both the validation and the practicability of the hypothesis in the real world. A testbed set-up is in fact an emulated small-scaled communication environment. A success of the suggested approach on the testbed will prove that the hypothesis can exist in a communication miniature and simultaneously increase the possibility that the hypothesis will work for real-world projects. Whereas a breakdown of the experiment may be due to many reasons: failure of the hypothesis, impracticability of the approach, or incompatibility of the approach and the testbed. We call this method *experiment approval*, or *emulation approval*.

Proposing the project to real telecommunication players, applying it to a real-world network, and collecting feedbacks from users will be the most practical and meaningful but most difficult method. Even though real engineering problems all need testing in this way, it is unrealistic to do so in a university environment. This method is called *user approval*.

Although the above four types of methodology are available for validating the hypothesis of constructing human-like intelligence embedded network, we select the first two methods for proof-of-concept purpose within the scope of the research. (1) The concept-interpretation method provides a theory basis for other three methods and is therefore the premise of using any other method. (2) The simulation-approval method is able to prove the feasibility of the concept in a simulation or software environment and is comparably easier to implement in university labs than the last two. Based on simulation results, one can make qualitative analysis to system functionality and behavior. Yet this method has limitation in providing practical data for industrial use. (3) The experiment-approval method is able to validate system performance and then provide more valuable, factual data for system operating. However, the method has a high demand on funding support and requires an much more advanced experimental environment than just a single person in university-lab environment. This method is therefore excluded from testifying the project results. (4) The last method is designed for and only used by network operators who maneuver the real network. Other than the above reasons, compared with pure science and engineering researches, the research on social networking has its intrinsic difficulties in collecting and providing accurate data because of the variety and dynamics of humans. In addition, it requires a more complex and considerate design to accommodate both machine factors (i.e., compatibility) and human factors (i.e., trust).

1.3.4 Concerns: Identity, Privacy, and Security

As users' human intelligence and social features step into communication process, several concerns need attention from network designers.

First, both the network and users need a few identify validations when performing human-like intelligence. With respect to the network, network components involved with human-like intelligence need to present themselves to network management in terms of usability and functionality. The network then needs to identify a service-session involver who is both technically and personally qualified for handling an on-going communication event. With respect to network users, only the privileged users who take the initiative to share their social network with others are qualified for the intelligence service when they demand one.

Second, the network needs to pay attention to the hidden troubles caused by user privacy. Because the network takes in user social relation in the choice of service-provision manner, it may let out a user's information through his/her related people without his/her own permission. Then it is necessary to obtain an authorization from users when establishing a mapping of their social-relation network and to confirm with them on when, how, and in what events to use their social relations.

Third, trust-related issues play an essential role when establishing user social networking in the communications network. Users' major concern with social communications network is the safety of leaving personal profiles (with social relations) on the network, because malicious resources may obtain users' personal information through their social relations. Hereof, mutual trust fidelity between two users fits in well to help determine the content and amount of information traded between the two users. Even if malicious resources have reached the entry of a user's profile through his/her related people, they are unable to access the content of the profile without passing the user's own trust verification.

Some of the above concerns will be addressed in future sections (i.e., trust) and some remain as open research topics on human-network interaction. We assume that these concerns have all been taken care of and only focus on the communication efficiency obtained from establishing human-like intelligence in the network.

1.4 Contribution of the Intelligence-Network Proposal to Communications

The proposal of a humanized network addresses the necessity and importance of enhancing the existing network with human-like intelligence. Users are still suffering from the impersonal network in many fields (section 1.1). It is therefore necessary to get rid of those unfriendly elements from the network and to personalize the network as users expect. Being aware of network customers' likes and dislikes, desires, and preferences is essential for the future development of modern communications. In a communication environment, end users start a value chain; service providers or network operators, manufacturers, and designers then follow in sequence [10]. Users' willingness towards purchasing network services will thus affect the profit gained by all the rest in this value chain.

Compared with other network designs, the intelligence-network design has its novel and specific properties of emulating users' invisible thinking and capabilities, not just mimicking their visible expressions or actions. This project aims at enabling the network with users' social features such as desires, attitudes, decisions, actions taken, and even emotions. Equipped with these social features, the network will be able to make decisions like a human being during a communication session. It will also be able to serve users like a good friend such as notifying users when errors are detected or suggesting solutions to avoid the confliction of traffic.

The proposal looks at all the running communication events in the network as a whole and obtains an optimal successful-execution rate of these events from a statistic point of view. By mapping network users' social relations onto the network, the communication events that ran independently before on the networks become relevant via the socially related event involvers[10]. That is, the output of one communication event is able to positively affect the execution and the consequent output of other events via mutual event participators.

1.5 Thesis Development

The project chooses to apply human-like intelligence to the communications network to relieve users of the heavy communication burden. The thesis then presents the essence of the project by listing in sequence the background, objectives and blue print, system design, software realization, and validation of the intelligence-enhanced-network proposal.

We first compare human-like intelligence with other types of intelligence to clarify the importance and uniqueness of the proposal and then prepare the background for the design and approach by studying the existing virtual-society solutions (Chapter 2). After having investigated

[10] Event involver, also called "event participator", is a network user who participates in a communication event as the initiator, the receiver, or the assistant communicating party.

the distress that users suffer from modern communication burdens, we propose to enhance the communications network with human-like intelligence so that the improved networks are able to serve users in a humanly way (Chapter 3). The user-centric specialty of the proposal produces the practical approach of letting the communications network mimic users' intellectual behaviors and employs their social relationships in a virtual-user system (Chapter 4). We then simulate the approach in a software environment to testify to the functionality of the proposal (Chapter 5). Final results from the software experiment will prove whether the proposal of applying intelligence to the network exists or not (Chapter 6) and, if it exists, how much positive improvement in performance the upgraded network can achieve (Chapter 7). These results help conclude on (1) whether the virtual-user system is a suitable approach for the intelligence-network proposal and (2) whether the proposal is able to positively affect network performance, resultantly reducing network users' communication burdens (Chapter 8).

Chapter 2 Viewing Existent Network Intelligence and Social Networking

People in the information age expect an easy and customized communication lifestyle. The realization of such lifestyle requires a comprehensive research into the overlapping fields of sociology, engineering, and computer science. The current technical realization of these overlapping can be summarized as in Figure 2-1:

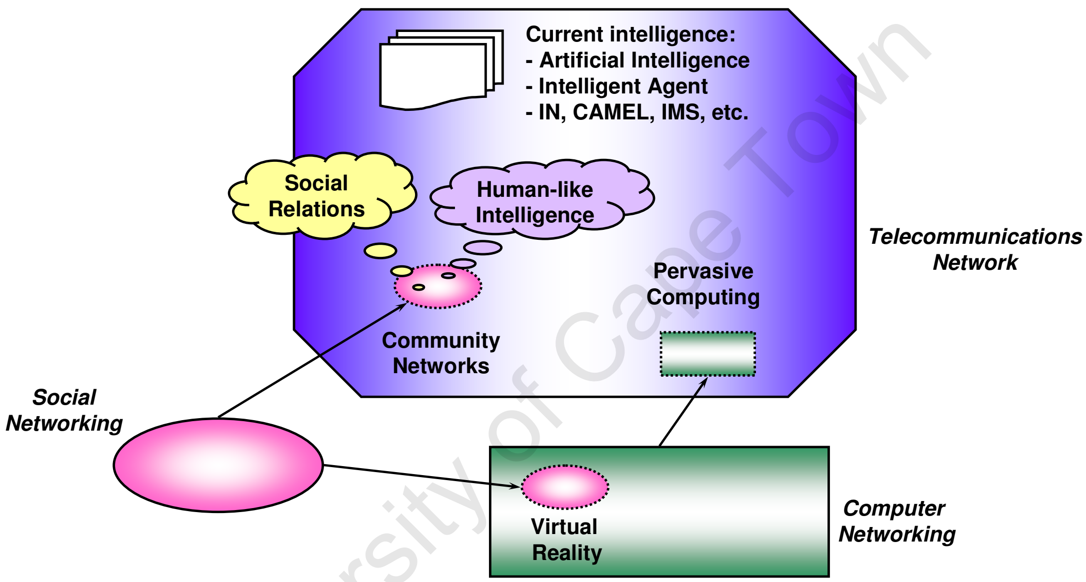


Figure 2-1. Composition of newly defined Heterogeneous Networking

Figure 2-1 illustrates the heterogeneous networking that contributes to humans' fabric of life. Here heterogeneous networking is not only referring to the commonly used notion of internetworking different networks that are using different protocols, but is referring to the collaboration of the different networking methods in social networking, telecommunications network, and computer networking. (1) Social networking uncovers how humans connect to each other and proposes solution for communications based on these connections [11] (section 2.1). (2) Telecommunications network transmits signals on radio frequencies over long distance for the purpose of communications. (3) Computer networking refers to the engineering discipline for the communications across interconnected computer systems or devices.

Incorporating social networking into computer networking is generally known as "virtual reality" (section 2.2.3) and widely applying computer networking to telecommunications network is called "pervasive computing" (section 2.2.2). "Community networks" provide an environment to timely and effectively set up a private network, which is also called community, to achieve better communications (section 2.2.1). However, the establishment of these communities is service-centered. It only follows certain rules preset by humans but lacks the intent to learn the rules from humans. Our research thus considers organizing communication-network nodes to interact with each other and with humans in the same way as humans socialize (i.e., mapping user "social relations"). The research involves mimicking "human-like intelligence" to accommodate humans social behaviors, personal desires, innovation skills[11] ([12]), and their adaptations in computing networks.

Existing research in telecommunications have already produced many advanced schemes and technologies to absorb part of human-like intelligence in network design. They use artificial intelligence to mimic human senses (section 2.3), set up intelligent agents to coordinate the users in the same community (section 2.4), and facilitate manipulating intelligence platforms to provide the users with user-friendly services (section 2.5). These schemes have improved network performance to a certain extent by respectively considering the sensorial, programmable, and applicable features of human-like intelligence. Yet there is room to improve each individual technology as well as coordinate these technologies by employing more human-like intelligence in the communications network.

2.1 Social Networking Analysis

"Social networking is built on the idea that there is a determinable structure with which people know each other, whether directly or indirectly" [13]. Early in the 1930s, people had

[11] Innovation skills refer to the types of skills that allow individuals to become innovative in what they do. They usually include cognitive skills (e.g., the ability to think creatively and critically), behavioral skills (e.g., the ability to solve problems and manage risk), functional skills (e.g., basic skills such as reading, writing, and numeracy), and technical skills (e.g., research techniques, project management, or engineering).

begun to study social members' relations. In describing these relations, they represented the members as dots and their relations as lines [14]. Meanwhile, some researchers began focusing on social groups within social systems such as family, work, associations, and clubs for easier analysis [15]. In 1960s and 1970s, people further explored the mathematical basis of social structure. They drew algebraic models of social cliques and established concepts such as the strength and distance of connections using theory and multidimensional scaling [15].

At present, a social network is generally defined as an interconnected system in which social nodes connect to each other by links with specific types of interdependency. These social nodes are such as individuals or organizations where the interdependency type can be value, financial exchange, or relationship. Social network analysis specifically looks at the social-relationship types of link between these nodes and works out feasible solutions to identify and solve real-world social problems [13]. For example, a reasoning agent capable of social network analysis can learn the social members' common ground from their social relationship and, based on such knowledge, make rules to assist them in achieving their common goals.

Here we describe several selected link metrics that are essential to a general analysis of social system ([16]) and are relevant to the project. We use Yang's simplified social topology in Figure 2-2 as an example to clarify these items.

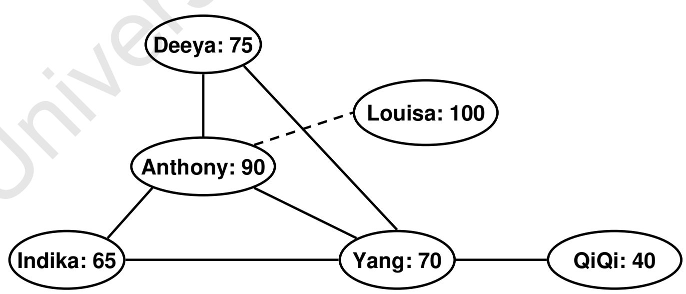


Figure 2-2. Yang's simplified social-relation topology.

Each user's absolute trustworthiness is marked beside his/her name as node weight.

(1) *Path Length* indicates the distance from one node to another and is generally the number of link hops from the former to the latter. For example, the path length from Yang to Louisa is "2". (2) *Betweenness* of a node indicates an average contribution of the node to the most efficient connections (i.e., the shortest path) between all pairs of nodes in the system, except

itself. For example, Yang's contribution to the successful establishment of all the most effective connections for Deeya, Indika, Anthony, Louisa, and QiQi's in percentage is respectively 50%, 50%, 25%, 25%, and 100%. Then Yang's betweenness in the above topology is "50%". (3) *Closeness* of a node indicates the extent to which the node is near all other nodes in the system and, therefore, is calculated as the inverse of its average shortest geodesic distance to any other node. For example, the shortest paths from Yang to Deeya, Indika, Anthony, Louisa, and QiQi are respectively 1, 1, 1, 2, and 1, so her closeness is about "83.33%". (4) *Eigenvector Centrality* of a node measures the importance of a node by summing up the weights of all its adjacent nodes [17]. For example, Yang's eigenvector is "440" if taking a node's absolute trustworthiness value as node weight. (4) *Centralization* of the system signifies the distribution of links on all nodes. (5) *Density* of the system indicates the connectedness of the system nodes using the ratio of the sum of all factual links to that of all potential links between any two nodes. For example, the number of all factual links is eight and that of all potential links is fifteen, and the density of the structure is therefore "53.33%". (6) *Structural Cohesion* of the system describes the number of essential nodes, whose leaving may cause a disconnection of the entire system, out of the number of total nodes. For example, the leaving of Yang or Anthony may cause the disconnection of QiQi or Louisa from the structure, and then the structural cohesion of the structure is "33.33%". (7) *Structural Equivalence* of the system refers to the extent to which a group of nodes have a common set of linkages to other nodes in the system. For example, both Deeya and Indika have common linkage to Yang and Anthony. (8) *Structural Hole* of the system refers to the newly added links that may link two previously unconnected nodes. For example, a direct link setup between Louisa and QiQi will facilitate their communication more efficiently.

2.1.1 Social-Network Examples on Internet

Social networking applies the social-network-analysis to the Internet to provide the users with novel social-relationship related services. A successful example of social networking - Facebook website - is shown in Figure 2-3:

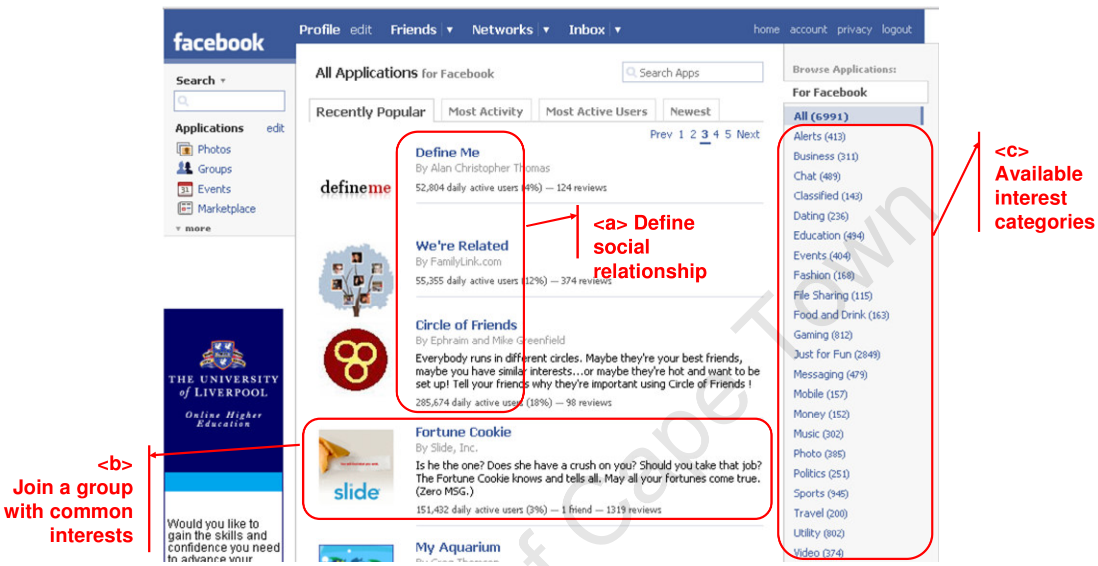


Figure 2-3. An application of social networking – Facebook.

Figure 2-3 prints the screen shot of an application webpage on the Facebook website under Yang's login. Facebook maintains an enormous database of network users' demographics, relationships, likes and dislikes, and expectations [18]. The users voluntarily offer their particular information for the purpose of advertising on and off the website. A user can define oneself and describe one's social circles such as relatives or friends (Figure 2-3a). Facebook can also broadcast a user's interests and obtain preferred information from the website by adding new applications under his/her login (Figure 2-3b). The user also has the flexibility to select events of interest from the web-provided list (Figure 2-3c).

Orkut is another outstanding social-networking site, which is operated by Google [19]. As the codes on its homepage declare, Orkut helps users "connect with friends and family using scraps and instant messaging," "discover new people through friends of friends and communities," and "share your videos, pictures, and passions all in one place." Orkut is similar to Facebook in that it allows a user to set up personal profile, invite friends to the site, join community, and deploy applications. By enabling all kinds of communication manners for the user to interact with friends on the website, Orkut encourages its users to expand the network, so that the website profits from the expansion.

2.1.2 Six Degrees of Separation

The theory of Six Degrees of Separation states that anyone on Earth can reach anyone else through a chain of acquaintances of no more than six people [20]. If a person is one step away from another person he/she knows and two-steps away from another person whom he/she does not know but his/her acquaintances know, then any person on Earth is no more than six-steps away from any other person. The theory indicates that people all over the world live in a "small world" as far as social relationships are concerned [21]. Social-networking researchers generally agree with such an assumption and have conducted several experiments to confirm it [22].

Applying the Six-Degrees-of-Separation theory to computer networks such as Internet, we can distribute information to users worldwide via the interlocking network. An example of this application is the LinkedIn website shown in Figure 2-4 [23].

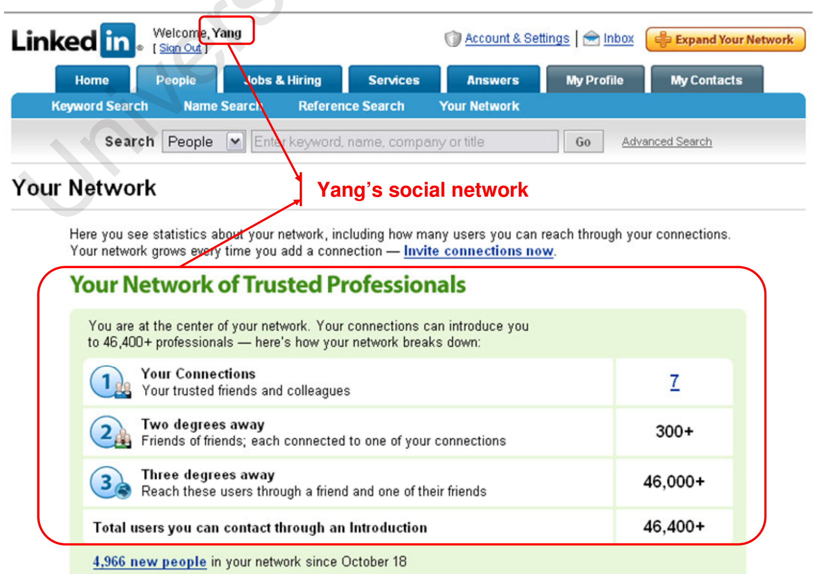


Figure 2-4. An application of social networking – LinkedIn.

Figure 2-4 explicitly shows the hierarchy of Yang's social relations. Initially, Yang only set the links to seven acquaintances and, therefore, she has a one-step or one-degree connection with each acquaintance. Then, through the seven acquaintances, Yang is exposed to more than 300 network users who are two-steps away from her and more than 46,000 users who are three-steps away from her. These potential connections create an opportunity for Yang to become the acquaintance of other users via her trusted users, leading to a safe expansion of her social network. Provided with both scalability and security, Internet service providers are able to distribute information via the interlocking network to users worldwide as well as confine the distribution in users' trusted social network.

2.1.3 Trust-related Issues as Link Metrics and Node Weights

Having adopted network users as the nodes and the users' social relationships as the links between the nodes, social network has two important concepts related to relationship. They are trust degree and trustworthiness.

We set up a communication scenario to explain the two concepts and their importance. To initiate a communication event between two social nodes, the source node needs to first establish a trust relationship with the target node. After investigating the relationship of the two nodes, the source node uses the investigation result to determine whether to and, if so, how to communicate with the target node. In this scenario, (1) the trust degree is the extent to which the source node trusts the target node in handling the event and (2) the trustworthiness is a node's intrinsic trust fidelity in processing any event from any other node. Both the source and target nodes have their respective trustworthiness values, but the value only works when the node receives an event from another node. The trust degree represents the source node's willingness for the target node to process an event and the target node's trustworthiness represents its reliability for handling the event.

In social network analysis, we represent the trust degree as the metric of links and the trustworthiness as the weight of nodes. Then the trust degree from one node to another only affects the information transmitted on the direct link between the two nodes whereas the trustworthiness of a node affects the information on all links directly connected to the node.

We then quantify the two abstract trust-related concepts in a proper manner so that they may be used the in computer or communication networks. Current researches on trust have provided a method of doing so by following the steps below. (1) Analyze the generic concepts of trust degree and trustworthiness. (2) Reinforce the basic concepts of trust degree and trustworthiness by considering as many causal factors of them as possible. (3) Formulate the trust degree and the trustworthiness. (4) Select suitable modelling languages for the implementation of the formulae. (5) Simulate trust degree and trustworthiness in software. Joining the above research methodologies into a job sequence enables us to define and calculate trust degree and trustworthiness.

2.2 Heterogeneous Networking

The three technologies that are stimulating information exchange are social networking, telecommunications network, and computer networking (Figure 2-1). Merging one technology with another produces hybrid computing products. These products not only inherit the features of respective technologies but also introduce new features owing to the cooperation of the technologies and are therefore able to stimulate the information exchange from a different angle.

Applying social networking to telecommunications network produces community network (section 2.2.1). Applying social networking to computer networking brings virtual reality (section 2.2.3). Combining computer networking with telecommunications network gives pervasive computing (section 2.2.2).

2.2.1 Community Network

Community network is a service-centric networking technology that enables a community of users to obtain services easily and freely. The community members generally have common service requirements, all contribute to the establishment the service, and may even be inhabiting in the same neighborhood. The provided services are generally specific functions, such as pure Wi-Fi access. The community networks prevail especially in hot spots or occasional get-togethers because these networks are fast to set up and are providing specific services.

Community network has many prototypes in the real world. The famous FON network entitles a community of people with free global WiFi access [24]. After having volunteered to share home wireless access, a FON community member is then able to access network through other users' shared network access points at other locations. This way, the user is able to use wireless access in a broader geographic area at a low cost. Another life example of community network is Boingo, which provides the community members with Internet access on diverse end devices (i.e., mobile phones and computers) [25]. Each member can use a universal identity to access the Boingo access points at those assigned hotspots.

However, community network so far is not perfect in integrating social networking with telecommunications so far. (1) Currently, a community network only provides a specific type of service and is restricted from extending to multiple functions. For example, FON has been set up only for easier access to wireless access in a wider area. If a FON community member has recently upgraded his/her shared access point with Bluetooth technology and he/she would like to share this technology with other members, he/she may encounter obstacles from FON. Either he/she has to go though a lengthy process of establishing agreement between him/her and FON network, or he/she is not allowed to do so because FON does not support the technology. (2) The establishment of user relations in a community is random and temporary. These relations cannot be used for other purpose of service provision in future. In the above-mentioned FON network, once a user leaves his/her community, he/she cannot keep the established social relations with other community members. When two users from the same community join another community, they need to set up a new relation between them in the new community. This activity not only slows down the process but also causes extra work for users. (3) At a time, a community only has a limited number of members to manage. In contrast, a real-world social network involves a worldwide group of users at any time because of its global connectivity (referring to the six-degree theory). Theoretically, it has a broader group of customers than a community when introducing a service. In conclusion, it is desirable that community networks permit flexible service-manipulating, establish fixed and broader range of user relations, and allow relationship transfer when needed.

2.2.2 Pervasive Computing

Pervasive computing, also known as ubiquitous computing, merges micro-electronic devices into humans' multitudes of life without being noticed by humans [26] [27]. These devices (i.e., microcomputers) collect realtime information of users from their persuasive aspects of life and provide constructive feedback when appropriate. They are supposed to be small-sized, mobile, programmable, and interactive.

Two major features of pervasive computing are extremely wide distribution and multi-functionality. First, pervasive computing spreads computer technologies in people's everyday activities. The novel applications cover much more than just erasable papers, wearable computers, electronic newspapers, and smart homes [28]. Second, mediated by pervasive computing, a single device can effectively operate multiple functions. One most promising instance of such devices is cell phone, which not only has a ubiquitous distribution among users but also supports the development of multi-functions [29]. A cell phone may be also used as transponder when passing a toll expressway, remote control for garage door and for TV, or heart-beat-rate monitor.

Despite the available theory and prototype of pervasive-computing instances such as electronic newspaper, their large-scale production will take time. In addition, pervasive computing is currently service-centred and not yet human-centric. Besides, each instance of pervasive computing implements a specific function and serves each user as an individual. It needs to consider more of users' social properties.

2.2.3 Virtual Reality

Virtual reality refers to the technologies that create a computer-generated environment for the humans to feel like being present in a real world [30]. It addresses entertaining humans' spirit world or training their skills with a high level of simulated sensory fidelity. To achieve such high level, virtual reality concerns the following. First, it simulates all human sensory (visual, auditoria, and other sensory cues) as closely as possible to what the users would experience in the real world. Second, virtual reality obtains and replays user memories to simulate their real-world experience, so that the users react the same way as in the past circumstances. This scenario is specifically for collecting users' real information. Third, virtual reality maps our social system (i.e., social relations) in the computer network to generate the same type of familiarity with social lives for users as what they are experiencing in the real world.

Virtual reality has successful applications in various fields such as gaming and tele-health. A good example is SecondLife, which establishes an online virtual society for users and presents them as avatars in the society [31]. In the virtual world, an avatar can set up "his/her" personal profiles (e.g., choose favourite clothes and uploading photos), buy and sell properties (i.e., land) to make money, and meet people. SecondLife was originally for entertainment, but some scientists have employed its platform to develop meaningful applications in tele-health field [32]. They create online-support sessions to guide psychological assessment and rehabilitation, illustrate physical therapy, and recommend appropriate clinics after having interacted with patients online. These applications serve especially the patients who have disabilities, are living far from clinics, or just seeking for follow-up studies.

Current approaches of virtual reality mainly apply to the computer world and to several specific fields such as gaming, telehealth, and training process (e.g., the training for fire escape). Embedding the strong "social" attribute of virtual reality in other fields such as communications network will open more room for technological development.

2.3 Sensorial Artificial Intelligence vs. Logical Human-like Intelligence

Artificial intelligence is the prime focus of general intelligence research. However, most researchers address programming the sensorial features of biological intelligence (section 2.3.1) and have not paid much attention to the logical features of them (section 2.3.2).

2.3.1 Definition, Features, Applications, and Scope of Artificial Intelligence

The term Artificial Intelligence (AI) was first coined in 1956 by John McCarthy as "the science and engineering of making intelligent machines" [33] and, nowadays, it generally refers to the computational and synthetic intelligence exhibited by an artificial entity such as a computer or a machine. Research in AI is concerned with producing such synthetic entities to automate tasks that require human intelligent behaviours. These tasks include controlling, planning and scheduling, diagnosing and answering questions, speaking, handwriting, and so on. The research has also created an engineering discipline that focuses on providing solutions to real-life problems through knowledge mining and software realization.

The AI applications overlap a majority of science fields [34]. They include the programs that (1) enable artificial devices to play games such as chess, (2) make decisions in real-life situations in expert systems[12] ([35]), (3) understand natural human languages, (4) simulate biological intelligence in neural networks, and (5) detect and react to sensory stimuli in robotics[13] ([36]). The communications network has especially absorbed several AI virtues in network operation, such as expert system for management and neural networks for algorithms.

One of the biggest difficulties with AI is that of comprehension. Currently, no devices are able to conduct full artificial intelligence and resultantly simulate human behavior to the full. (1) Although the International Business Machines Corporation (IBM[14] [37]) super-computer – Deep Blue – has defeated the world chess champion Gary Kasparov in a chess match in May 1997 [38], the majority of games are still under the control of humans and the computers continue to lose to them. (2) The expensive AI-embedded expert systems are only applicable to specific fields such as that of medicine and engineering in special situations. (3) The programmed translation system

---

[12] Expert system, also called knowledge-based system, is a computer program that contains the knowledge of specific subjects and human experts. It utilizes what appear to be reasoning capabilities to reach conclusions. Its most common form is a program that provides mathematical analysis to a class of user-supplied problems and, based on the preset rules and design, recommends a set of user actions to implement the solutions to the problems.

[13] Robotics is "the science or study of the technology associated with the design, fabrication, theory, and application of robots".

[14] IBM is an international computer technology and consulting corporation headquartered in New York, United States of America. IBM manufactures and sells computer hardware, software, and infrastructure/hosting/consulting services in areas ranging from mainframe computer to nanotechnology.

of natural human languages can neither provide clear human voices nor accurately identify them in some scenarios. The voice recognition systems that convert spoken languages into written words simply take dictations yet do not understand what they are writing. (4) Although the AI researches in neural networks have made certain progress in the recognition of face, speech, and action, they are still far from commercialization. (5) The AI-enhanced computers in robotics are capable of a limited number of tasks. They have great difficulty in identifying objects based on appearance or touch and they handle objects clumsily.

2.3.2 Need to Enrich Artificial Intelligence with Human Logic and Abilities

The above analysis discloses that most on-going research on AI only focus on the sensorial features of human-like intelligence and not on the logic behind human-like intelligence behaviors. These sensorial features have only respectively described human behaviors in various fields such as pervasive computing (e.g., gaming), ambient intelligence (e.g., recognition of human senses), and mechanical analogy (e.g., robotics). The logic feature further includes why humans behave in certain manners and how they conduct a pattern of such behaviors in a systematic way. Therefore, it is very necessary for the AI entities to understand the logic behind human behaviors. With such understanding, the entities are able to perform tasks on their own, independent of human instruction. In addition, the existent base abilities to handle the sensorial features may facilitate the acquirement of the abilities to handle the logic feature in AI entities.

In addition, there is no practical AI application in the field of telecommunications at present. However, using artificial entities to represent users in the telecommunications network may greatly facilitate the process of communication sessions. Knowing users' preferences and session-processing habits respectively via the social and logic features, these entities are able to process sessions according to user desires as well as suggest better solutions when necessary.

An approach for importing the logic-enhanced AI entities into telecommunications is to implement the needed intelligence as an expert system specifically for the communications network. The system first collects network users' communication behaviors in real-time. It then uses the collected information together with the preset rules (a reflection of the session-processing logic) to make decisions on factual session processes. The system differs from other expert systems in the ability to obtain real-time information from users for decision-making, other than just following the preset instructions from human experts.

Various types of expert systems are available for developing an AI application in the real world. They include the bio-inspired or bio-mimetic, social or organizational-based, algorithm-based, cognitive, logic-based, knowledge-based, and hybrid systems [39]. For the dynamic communications network, the social logic-based expert system is an excellent selection.

Partridge and Hayes-Roth have built and agreed on a useful data flow to implement a social knowledge in an expert system, as shown in Figure 2-5 [40] [41]. The flow also prepares a methodology for importing human-like intelligence into the communications network.

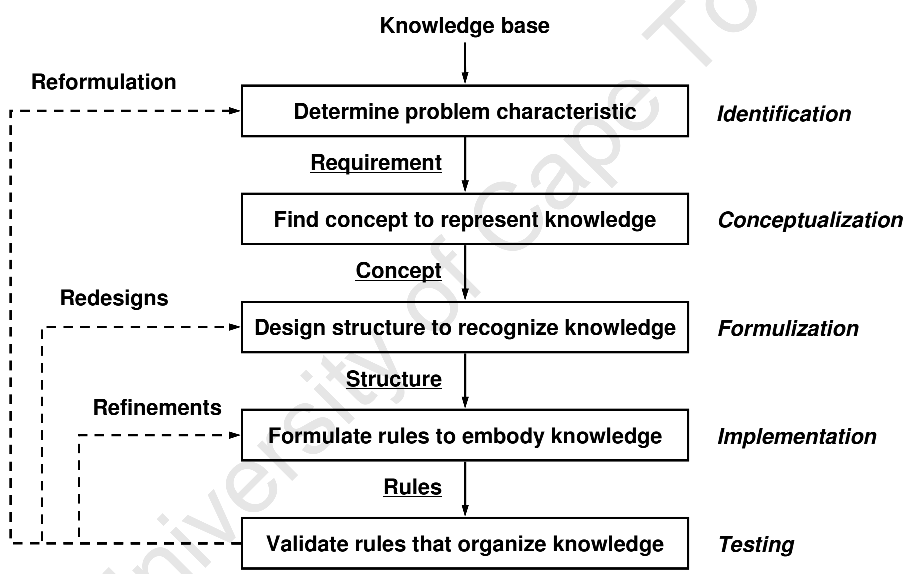


Figure 2-5. Control flow of Expert System.

(Modified from the figure in [40])

The data flow in Figure 2-5 addresses the point of controlling an expert system and this control methodology can be widely applicable in a variety of fields such as communications. Based on the identified scope and specific features of an observed real-world problem (stage *Identification*), the system first develops the elementary concept of the problem using selected key patterns (stage *Conceptualization*). It then breaks the concept into several independent function modules, identifies their mutual relations, and applies to them the strategies of operation and management (stage *Formulization*). By transforming these actions into referable operating rules and implementing these rules in software design, an executable system comes into being (stage *Implementation*). The last stage concerns validating the functionality and evaluating the performance of the prototype system (stage *Testing*). Using the output of the test as part of the input for most early stages facilitates refinery of the target system. Through the refining process, the system can closely simulate the nature of the real-world problem and propose the most appropriate solutions (referring to the dashed arrows).

Once the system has been set up, it runs endlessly for two reasons. First, the knowledge base is never complete. There is always the possibility of adding more knowledge to improve system performance or broaden system scope. Second, there is often no absolute correct solution for the system to produce. There is thus always the possibility of adding (or improving existing) heuristics to achieve better results, more explicit diagnoses, and more approximate predictions.

2.4 Intelligent Personal-assistant Agent vs. User Assistant with Human-like Intelligence

To implement biological intelligence in the mechanical world, one feasible approach is to let intelligent agents act on behalf of humans in terms of themselves as individuals, the features they possess, or the tasks they perform (section 2.4.1). The future study of agent systems will bestride the disciplinary boundaries of the existent ones by focusing on the society-, culture-, and communications-aspects that emerge from the interaction of the agents (section 2.4.2).

2.4.1 Definition, Features, Applications, and Scope of Intelligent Agent

Generally speaking, an Intelligent Agent (IA) is an autonomous system that senses the surrounding environment and performs actions over time according to the design objectives, expecting to eventually benefit the users [42] [43].

A combination of four properties differentiates intelligent agents from non-intelligent autonomous systems and software programs [42]. (1) Autonomous. An IA takes the initiative to detect the change of external environment and has control over its own actions. Besides following a set of preset behavior-controlling strategies, an IA has the freedom to take action and respond the way it feels appropriate during its communication with other entities. (2) Environment-specific. An IA interacts with a specific external environment. Once away from that environment, the agent cannot perform the same tasks as it did, resultantly loosing the role it played in that environment. (3) Reactive and proactive. An IA is able to initiatively sense the changes of its situated environment and timely respond to them using its own algorithms. That is, the agents proactively refine the algorithms that are independent of delegated tasks and reactively perform the tasks according to the algorithms. (4) Continuous. An IA keeps on sensing and acting on the environment according to its own agenda.

In addition to the above compulsory properties, the following four features contribute to the speciality of different types of IA. (1) Communicative. An IA communicates with other agents in a mutually understanding way. These communication actions are to assist these agents with their respective tasks. (2) Adaptive. An IA is aware of an emergency situation and is able to adapt accordingly. The agent first leans from its previous experience and, based on which, anticipates future situations. It then uses its memories and anticipations to adjust its future behaviors. (3) Mobile. An IA in the form of a software program is transportable between machines in terms of physical implementation and between platforms in terms of functional implementation. (4) Flexible. An IA's actions depend on its factual situation – the design objectives and the real-time external environment – not on scripts.

There are many ways to classify IAs – by situated environments, by possessed features, or by performed tasks. Figure 2-6 presents a reasonable way of classifying these patterns.

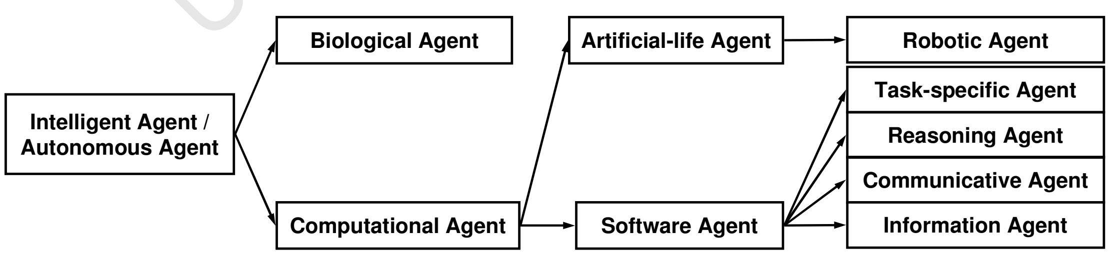


Figure 2-6. One way to classify intelligent agents.

(Modified from the figure in [42])

The classification in Figure 2-6 organizes the hierarchy of IA categories using our general understanding of intelligence. In a general sense, we consider an entity that acts autonomously as an intelligent entity. The intelligent entity can either be biologically intelligent (*Biological Agent*) such as a human being in the real world, or be computationally intelligent (*Computational Agent*) such as a piece of computer program that implements the intelligence characteristics of human beings. Technically speaking, this type of differentiation between agents originates from a consideration of their respective situated environments. Then according to the degree of intelligence that IAs exhibit, a Computational Agent can be either an *Artificial Agent* that is capable of perception and action by following its own agenda or a *Software Agent* that implements the intelligence abilities of adapting and learning in software programs. The latter can be further categorized into four types of agents by the tasks to be performed. A *Task-specific Agent* is dedicated to a specific task, a *Reasoning Agent* interprets real-world concepts and determines actions based on the analysis of the concepts, a *Communicative Agent* communicates with other agents such as human beings, and an *Information Agent* coordinates information.

The core technologies of the IA have stimulated a wide range of applications, especially in emerging electronic commerce[15] ([44]), information economics[16] ([45]), and Internet [46]. The corresponding products span data mining, knowledge sharing, network management, information integration, software-mobility solution, web browsing, personal assistant, and so on.

The widespread dissemination of IA products in the social, economic, and other fields of humans has enabled all users to access a great number of technical resources that require advanced skills. The users will then have to digest large amounts of information and be engaged in several different activities at the same time. Here, instead of providing the users with the

[15] Electronic commerce, often spelled as "e-commerce", refers to an economic system that consists of buying and selling products or services over electronic systems such as the Internet and other computer networks.

[16] Information economics is a branch of microeconomic theory that studies how information goods or information services affect the economy and economic decision-making.

convenience to access a pile of information, the IAs have imposed extra work for them. Figure 2-7 illustrates this type of inconvenience.

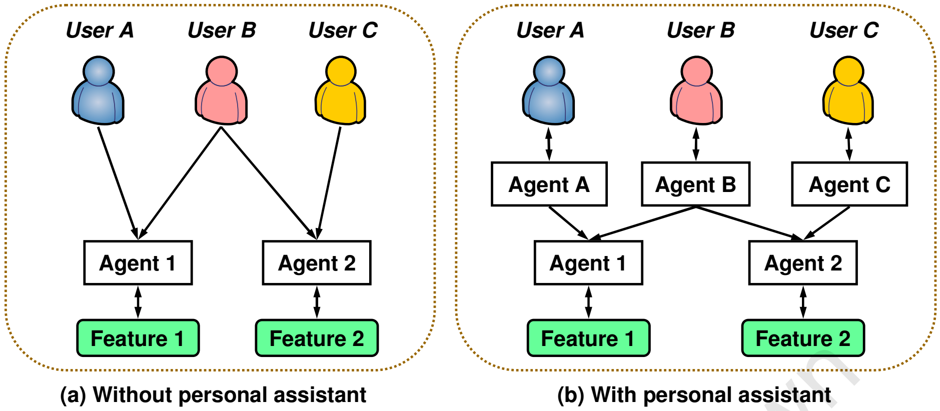


Figure 2-7. General structure of a multi-agent system between users and features.

Agent 1 and Agent 2 in Figure 2-7 each provide one feature for users. In Figure 2-7 (a), if User B would like to use both features, it has to make the effort first to learn that Agent 1 offers Feature 1 and Agent 2 offers Feature 2 and then to use Agents 1 and 2 for desired features. One prevalent solution for waiving the learning work is to assign each user a user-specific agent as in Figure 2-7 (b) [47]. Each user-specific agent basically plays a role as personal assistant. User B only tells its assistant Agent B of what features it wants and Agent B will go and find out what lower-level agents to use. Although the multi-agent system gets more complicated with the involvement of three personal-assistant agents, the users' work of choosing agents is saved.

2.4.2 Need to Enhance Intelligent Agent with Social Properties

Although the system has imported user-specific agents to intensively assist users with their work (section 2.4.1), it is still far from satisfying their desires for emergency services and new features. In Figure 2-7 (b), User A has no way to use Feature 2 if Agent A has no information of Agent 2; and similar for User C with Feature 1.

However, if these user-specific agents relate to each other in a certain manner, a user may be able to gain access to originally unavailable services and features via the user-specific agents of other users. Figure 2-8 provides an illustration on using user relationships to adjoin their user-specific agents.

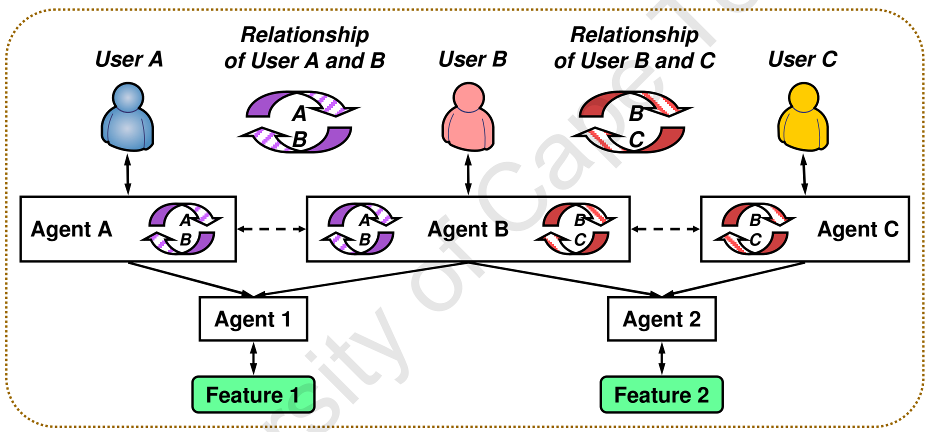


Figure 2-8. Multi-agent system with social-feature enhanced personal assistant.

Information in Figure 2-8 illustrates an approach of connecting two user-specific agents by the relationship of their represented users. For example, both Agent A and Agent B have kept an image of the Relationship of User A and B. In this way, a connection has been set up between the two agents. Then, in theory, User A is able to gain access to Feature 2 by following the route of User A, Agent A, Agent B, Agent 2, and Feature 2.

Implementing user relationships on the user-specific agents in a multi-agent system possesses two features. Firstly, it establishes a cooperative relationship between the users via their correlated user-specific agents. Secondly, it extends the availability of service features. We take the users, the user-specific agents, the feature agents, and the features in order as four sequential layers of a hierarchy. In Figure 2-7, the procedure of a user fetching a feature goes vertically through the hierarchy and, resultantly, the user is restricted in using all features. However, after the set-up of the relationships between the user-specific agents (Figure 2-8), the procedure is able to flow not only vertically between layers but also horizontally within the user-specific-agent layer. In this way, each user has the opportunity to access more features.

The approach of importing user relationships via their user-specific agents is predictably feasible because users all socially relate to each other. According to the theory of Six Degrees of Separation, if a person is one-step away from another one whom he or she knows and two-steps away from another one who is known by his or her acquaintances, the person is no more than six-steps away from any other person on Earth [48]. Supposing this theory is true in sociology, any user in Figure 2-8 is then able to use any available feature through its relationships. The approach is applicable in the communications environment because, in most cases, the users who communicate with each other always have a certain social relationship. Therefore, they are able to and would like to share and help transfer the services and features.

2.5 Intelligent Architectures vs. Social Features

A variety of current intelligent architectures provide value-added services to users in their respective carrier communication networks (section 2.5.1). However, they generally lack adequate consideration of users' social features (section 2.5.2).

2.5.1 Pros and Cons of Typical Existing Intelligent Architectures

Over the last two decades or so, the communications network has been cooperating with several types of intelligent architecture such as the Intelligent Network (section 2.5.1.1), the Customized Applications for Mobile network Enhanced Logic (section 2.5.1.2), and the Internet Protocol (IP[17], [49]) Multimedia Subsystem (section 2.5.1.3).

---

[17] IP is a data-oriented protocol used for communicating data across packet-switched networks.

2.5.1.1 Intelligent Network Capability Set 1/2/3/4

Intelligent Network (IN) is a network architecture that allows network operators to provide value-added services in addition to the standard telecommunication services. The creation, management, and delivery of these value-added services are all independent of the underlying networks. Initially, IN is applicable to all types of fixed telecommunication networks such as Public Switching Telecommunication Network (PSTN[18] [50]), Packet-Switched Packet Data Network (PSPDN[19] [51]), and Integrated Service Data Network (ISDN[20] [52]).

The generally accepted IN Conceptual Model (INCM) from the Consultative Committee for International Telegraphy and Telephony (CCITT[21] [53]) is a four-plane architecture framework as shown in Figure 2-9 [54]:

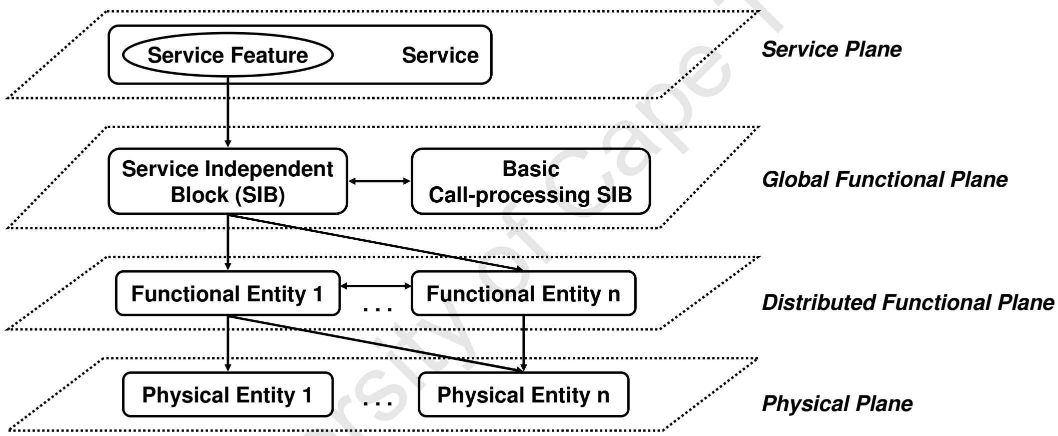


Figure 2-9. Intelligent-Network conceptual model suggested by CCITT.

[18] PSTN generally refers to the variety of circuit-switched analog telephone networks and services.

[19] PSPDN generally refers to a network that provides data services by using packet switching technology, commonly supporting X.25 packet-switching interface.

[20] ISDN is a Wide-Area-Network oriented data communication service that is capable of simultaneously transmitting a range of services such as voice, data, and video.

[21] CCITT (now ITU-T) was a telecommunications organization that recommended worldwide standards for common carrier communication services concerning technical, operational, and tariff-related issues.

(Modified from the figure in [54])

The simplified version of INCM depicted in Figure 2-9 comprises four planes. In the service plane, a service is characterized by several service features. In the global functional plane, a service feature is realized via the interworking between a relevant service independent building block (SIB) and the basic call-processing SIB. In the distributed functional plan, each SIB is realized through a sequence of particular service-feature actions in their situated functional entities. In the physical plane, these functional features are realized by several physical entities.

The IN Capability Sets (IN-CS), i.e., network capabilities that the IN architecture can provide, evolve in four stages [54]-[58]. IN-CS1 defines two types of basic intelligent services, essential network functions, control principles, feature interactions, and the functional relationships and interfaces of elements. Based on IN-CS1, IN-CS2 appropriately categorizes the intelligent services into telecommunication services, service-management services, and service-creation services. It further sets up detailed control principles according to the types of newly added services. IN-CS3 adds to the ability to support mobility and IP. IN-CS4 applies IN services to Voice over IP (VoIP[22] [59]) as supplementary services.

Although, the IN architecture is expected to provide new services to all types of networks, its practical realization (i.e., IN-CS1/2/3/4) lacks the flexibility to do so. Firstly, IN-CS1/2/3/4 cannot completely separate the operation of service processing from that of call processing. It only takes the value-added services as a supplementary function to the basic call services and occasionally activates them at trigger points. Secondly, IN-CS1/2/3/4 is, in fact, only applicable to the fixed network for the time being. Lastly, it is only supported in a single vendor environment even in theory.

For differentiation, we refer to the above IN as an "Intelligent Network" whereas the network with human-like intelligence embedded as an "Intelligence Network" in what follows.

[22] VoIP is the routing of voice conversations over the Internet or through any other IP-based network.

2.5.1.2 Customized Applications for Mobile network Enhanced Logic

Similar to the act of applying the IN concept in the fixed network, 3GPP adopts the Customized Applications for Mobile network Enhanced Logic (CAMEL) technology to apply value-added services in a multi-vendor mobile environment. The environment incorporates both circuit-switched (CS[23] [1]) and packet-switched (PS[24] [1]) networks, such as a combination of the Global System for Mobile communications (GSM[25] [60]) and the General Packet Radio Service (GPRS[26] [61]).

Figure 2-10 depicts the major functional entities of a phase-4 3GPP CAMEL architecture:

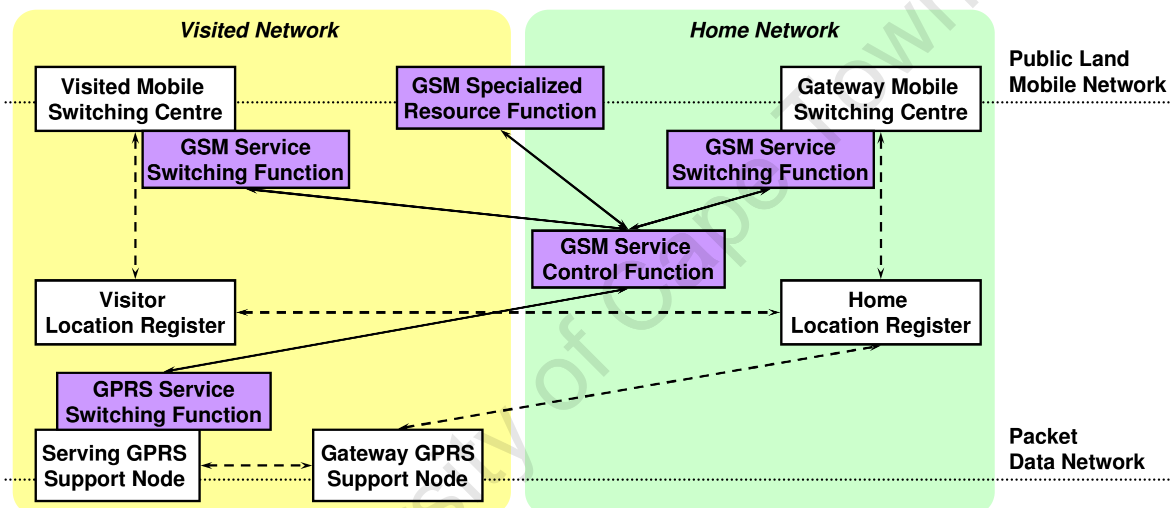


Figure 2-10. Customized-Application-Mobile-network-Enhanced-Logic structure by 3GPP.

CAMEL enhanced-logic nodes are accented in filled rectangles. (Modified from the figures in [62][63])

[23] CS is a switching technique that establishes a dedicated and uninterrupted connection between the end systems for the duration of a communication session.

[24] PS is a switching technique that divides the transferred data into packets and forwards these packets across the network along a route according to the information contained in their header.

[25] GSM is a second-generation cellular system transferring digital information in both signaling and speech channels. It has introduced great enhancements in security, quality, and ability to support integrated services.

[26] GPRS is a Mobile Data Service available to GSM users and IS-136 users.

Figure 2-10 illustrates how CAMEL provides value-added services in a multi-vendor mobile environment. The environment comprises Public Land Mobile Network (PLMN[27] [64]) and PDN, respectively represented by GSM and GPRS. We distribute CAMEL intelligence logic in the environment by placing the CAMEL nodes on the existent GSM and GPRS nodes.

There are four phases of CAMEL evolvement. CAMEL phase-1 sets up basic call-control functions for all GSM calls. CAMEL phase-2 realizes the function of cost charging in and allows direct announcement from the visited PLMN. CAMEL phase-3 becomes capable of handling GPRS sessions, performance monitoring, mobility management, Short Message Service (SMS[28] [65]), and multiple ways of charging. The most recent CAMEL phase-4 possesses the functions of handling multi-legged calls and initiating rich GSM calls[29] and is able to work with IMS.

Despite these advantageous features, the deployment of CAMEL is costly due to the frequent interworking between the enhanced functional entities (Figure 2-10). Furthermore, because CAMEL in visited networks is supposed to operate the same as that in home networks, the usage of CAMEL is therefore limited to a small area. Other than these problems, CAMEL inherits legacy IN problems, such as incomplete separation between call processing and service processing.

Noticeably, there is another intelligent architecture Wireless Intelligent Network (WIN) that plays an equal function as CAMEL but on a different type of wireless network – ANSI-41[30] ([66]) based wireless networks. The WIN concept is developed by the Telecommunications

---

[27] PLMN refers to all mobile wireless networks that use land based radio transmitters or base stations.

[28] SMS is a service that allows a short message entity to send short text messages to another entity.

[29] A rich GSM call refers to a multi-function service provided by GSM network, such as a combination of a voice call, a SMS, a video downloading, and a file sharing.

[30] ANSI-41 is a network standard that identifies and authenticates users, and route calls on mobile phone networks by allowing information exchange on switches.

Industry Association (TIA[31] [67]) WIN Task Group (TR-45.2.2.4). The service-independent WIN architecture separates the service logic from the wireless network switch and implements the logic as feature functionality on other intelligence-related platforms in the network. This way, WIN provides users with the capabilities for rapid service deployment, service creation, and service customization.

2.5.1.3 IP Multimedia Subsystem

With the fast convergence of fixed and wireless networks, 3GPP and The Internet Engineering Task Force (IETF[32] [68]) advocate an IP Multimedia System to provide both fixed and mobile users with multimedia services through Internet-based protocols [69]. A conceptual IMS architecture is presented in Figure 2-11 [70][71]:

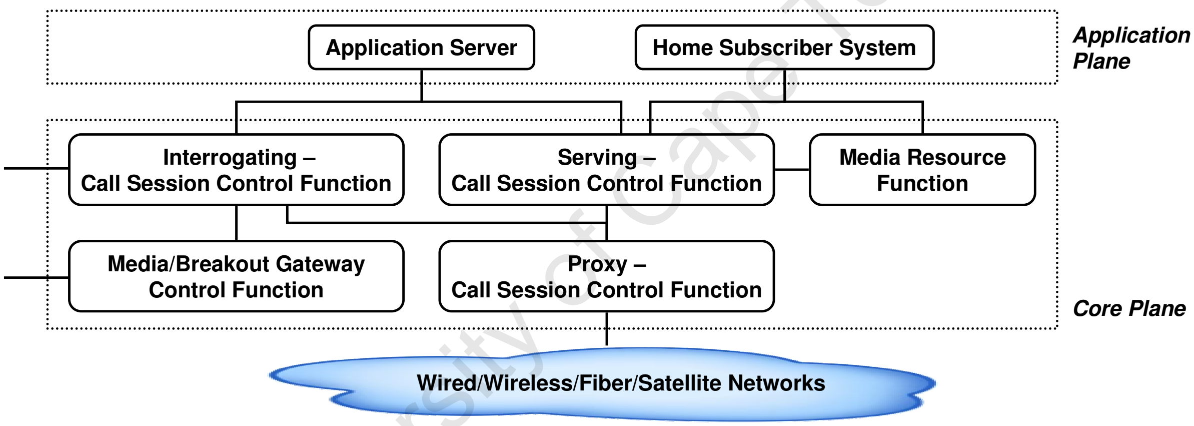


Figure 2-11. IP-Multimedia-Subsystem architecture suggested by 3GPP.

(Modified from the figures in [70][71])

[31] TIA is a trade association that delegates global ICT industries in many ways and especially enhances the business environment for telecommunications companies.

[32] IETF is a large open international community concerned with the evolution and smooth operation of the Internet. It is also in charge of developing Internet standards.

The IMS architecture, as illustrated in Figure 2-11, mainly acts as a middleware[33] ([72]) layer between users and networks. It upwardly provides independent service platforms for service developers to develop innovative services and provides databases for users to store their communication information, respectively such as Application Server (AS) and Home Subscriber Subsystem (HSS). It is downwardly accessible to all types of underlying access networks. The IMS core mainly responds to registration and session establishment using Session Initiation Protocol (SIP[34] [73]) and interacts with other CS- or PS-data networks.

The above three intelligent architectures address developing an environment to provide value-added services in their respective served networks. However, these services are only supplementary to the existing telecommunication services. That is, the network operators are only interested in enhancing the capacity of their networks to meet users' diverse service requirements. They have not considered establishing communications from the user side. A user has no right to request the network to provide whatever services he/she wants anywhere, anytime, and on any device. Instead, the user has to try all means to complete a session to suit the network's status.

2.5.2 Need to Import Social Features into Intelligent Networks

As has been mentioned above, current network intelligence architectures provide value-added services more from network operator side. The operators are currently concerned more about whether they can offer technically satisfactory sessions to a single user. However, they have ignored the enormous effect of increasing interactions between multiple users on their communications. It is certainly necessary and beneficial to establish communication according to users' interactive behaviors due to the following reasons.

---

[33] Middleware is software that mediates between two disparate application programs across diverse computing platforms and networks.

[34] SIP is an application-layer signalling protocol for creating, modifying, and terminating sessions with one or more participants.

Firstly, the two communicating parties in a session are certainly connected in a social relationship. With respect to personal communications, a user uploads or obtains information from some network resources maintained by another user who has the same interests. With respect to general communications, only the users in a certain relationship require the exchange of information through the network. With respect to group communications, communication streams only occur between users with common interests or those in the same social circle. Because all the above types of communication involve socially related users, it is a necessity to consider reconstructing the current network by absorbing the users' social relationships.

Secondly, a user's social circle enlarges with its accessibility to a broader range of networks. With the convergence of fixed and wireless networks, users are increasingly available for all sorts of communication services through a variety of handy devices, regardless of their diverse life backgrounds and different physical locations. The users thus get the opportunity to know more people with the same interests.

Thirdly, taking into account user lifestyle in communications enables the network to better serve the users. A user's communication life comprises its communicating activities in all social aspects such as banking, shopping, and online booking. Thus, with the inclusion of users' social features in communications, the network is supposed to be entirely user-oriented.

After having discussed the possibilities, necessities, and benefits of embedding user social features in the communications network, we then list the steps of doing so. Firstly, we characterize when, where, and how the users make their connections, how long the connections exist, and in what ways these connections serve the users in their communications. We then examine the ties between user communication activity and social activity, resultantly identifying how their sociality affects their face-to-face interactions in communications. Lastly, we identify how the users express themselves (especially their social features) through their profiles on the network. Altogether, the way to build a social communications network is to add the social features of the users into the existent communications network.

2.6 Summary of Chapter 2

This chapter first analyzes the properties of social networking and finds that many social features of users can be used to improve the performance of their communications. For example, we can use Six Degrees of Separation to identify trustworthy users for communications and use trust degree and trustworthiness to determine the appropriate communication manners. It then gives a bird's view to three types of heterogeneous networking (respectively community network, pervasive computing, and virtual reality) on their advantages and disadvantages.

This chapter further reviews the existent network intelligence and analyzes its advantages and disadvantages. Artificial-Intelligence technology provides the computer-based network with a strong capability to mimic human behaviors, but its application in other areas, such as the communications network, has been ignored. The Personal-Assistant-Intelligent-Agent technology is restricted of availing all services and features to a user due to its incomplete design of multi-agent system. The available intelligent architectures only provide value-added services and have ignored a group of related users' common needs in their structural designs. Overall, the communications network needs to import a type of intelligence that seriously considers users' social features.

With a need for the convergence of the communications network and the social network, we hypothesize to apply human-like intelligence as a representative of the social network to the communications network (Chapter 3). After studying the need for and the feasibility of importing human social characteristics into the current communications network, we will then propose a practical functional system to implement the hypothesis (Chapter 4 to Chapter 7).

Chapter 3 Proposing Human-Like Intelligence Enhanced Future Generation Network

The developing communications industry has brought a volume of benefits into communication users' daily life. Many novel and advanced technologies have especially considered user-centered factors into network design. Some technical schemes mimic human senses (e.g., artificial intelligence). Some set up user-like agents to correlate between users and services (e.g., intelligent-agent technologies). Some aim to provide users with more user-friendly services (e.g., existing intelligence platforms). Some organizes users in a community according to their common communication requirements (e.g., community network). Some address interconnecting physical network components in the way that humans normally socialize with each other (e.g., virtual society). These network components include personal computers, landline and cell phones, etc. All these schemes are concerned about copying human sensory features and its social-issue criterion to the network so that the network can improve performance and consequently serve users more efficiently and more friendly. Nevertheless, users still have to spend time and energy using services that the network offers. In fact, the more efficiently the network performs, the more communication events may occur within a fixed period and the quicker users have to respond to these events. Other than enjoying a multitude of new services, user life becomes more hassled.

We therefore propose to (1) copy user capabilities to handle communication events and (2) transfer a part of user communication activities onto the physical network. This way, the network is not only able to intelligently process communication events according to users' desires and preferences, but also able to take over part of the communication load for users. To evaluate and realize such a blue print, it is necessary to establish (1) a need for the current network to include human nature and (2) what aspects of human nature the network needs to import to enhance its capabilities (section 3.1). A detailed proposal follows below addressing how the network recruits human nature by defining and locating human-like intelligence in the network structure and by listing the expected mechanisms in its implementations (section 3.2). To support the proposal, we set up a virtual-user model and then address the parameter definition, architecture, scenarios, and features of the model (section 3.3). Finally, several assumed prospective life cases are narrated to help understand the major mechanisms needed to implement human-like intelligence in the network (section 3.4).

3.1 Expectation of Human Nature in Next Generation Network

Since the 1990s, Next Generation Network (NGN) has been one of the most talked about type of network that intends to accommodate a variety of personalized services over heterogeneous access networks [74]. Given the inherent need for diverse user-centric communications in NGN and easy communications between people, it is theoretically as well as practically reasonable to use NGN as the carrier of human-like intelligence, without losing the generality of applying human-like intelligence to the communications network.

We first have a brief overview of the near-future NGN communication environment (section 3.1.1) and then explicitly define the term "human-like intelligence" in the communication world (section 3.1.2). After determining what aspects of human nature are missing in NGN (section 3.1.3), we identify those most related to intelligence, interpret the chosen cognitive nature, and convert them into quantifiable network characteristics (section 3.1.4).

3.1.1 Nuts-and-Bolts Description of Next Generation Network

NGN is generally considered as a big IP-based network with open architecture, separate layers, clear interfaces, and excellent network management. The design of NGN aims (1) to seamlessly incorporate all the existing access networks and (2) to ensure all communication users enjoy any type of communication service anywhere, anytime, via any access terminal, with a good Quality of Service (QoS[35] [75] [76]), and at an acceptable price.

[35] QoS refers to resource-reservation control mechanisms that guarantee a certain level of performance to a data flow in accordance with the requests from application programs or the Internet Service Providers' policies.

International Telecommunication Union-Telecommunication (ITU-T [36] [53]) Special Group 13 has designed a very interesting NGN architecture that depicts NGN as four cascading function-independent levels distributed in two panels, as shown in Figure 3-1:

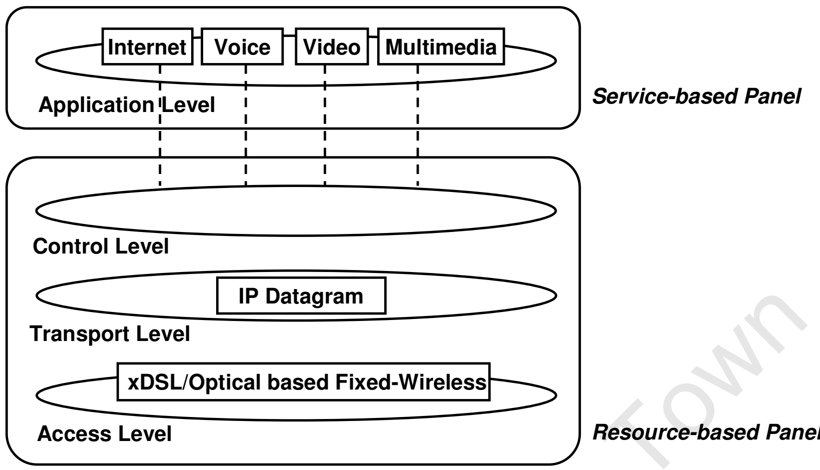


Figure 3-1. Next-Generation-Network architecture from ITU-T SG13.

(Modified from a figure in [77])

The NGN hierarchy in Figure 3-1 comprises four levels: application level, control level, transport level, and access level. The application level manages NGN services and provides development platforms to external parties, i.e., software developer. The control level provides interfaces for the upper level and is in charge of signaling to the lower transport level, which carries service data in IP packets. The access level at the bottom of the hierarchy provides open interfaces for existing access networks to access the NGN core. These four levels sit in two separate panels. The service-based panel has its unique element – the application level – to process all types of abstract network service. The resource-based panel has the control level, the transport level, and the access level to cooperatively distribute, utilize, and manage the physical network resources.

[36] ITU-T is an international organization that develops worldwide standards for telecommunications technologies.

This NGN-architecture design has explicitly distinguished between service functions and service realization and further separated them into the service-based and resource-based panels. This increased attention to service functions facilitates the formation of a flexible service-creating environment that provides an operable platform for service developers. With the service-function emphasized platform, developers need not worry about a service's physical realization (e.g., setting up the connection for a service session or maintaining service performance) while focusing on what the service can do for users.

3.1.2 Human-Like Intelligence in Communications Network

Human-like intelligence is semantically defined as the capabilities of humans to identify new problems and analyze them, to solve problems, and to contribute valuable inventions and services to the surroundings [7] [8]. To modulate and simulate human-like intelligence in the technical world, we develop the semantic definition as eight computer-recognizable human-like abilities: (1) recognition, (2) analysis, (3) planning, (4) troubleshooting, (5) abstract ideation, (6) expression, (7) languages, and (8) learning. That is, to function with intelligence as a human being, a network needs to be able to

(1) detect and recognize the presence of communication problems,

(2) analyze the types and the characteristics of the problems,

(3) carry out potential problem solutions,

(4) solve the problems by making proper judgments according to certain rules,

(5) foresee the potential problems that the communication sessions may come across,

(6) describe and periodically report the status of communication sessions to the network administrator,

(7) communicate in one or many universally acceptable languages, and

(8) learn the process methods from previous successful service sessions.

The above human-like intelligence has provided a logical procedure of processing an event (abilities 1-4) and has summarized the required attributes related to the procedure (abilities

5-8). If the network is able to follow such logical procedure to handle communications and to apply those attributes when necessary, it then has obtained the essence of human-like intelligence.

3.1.3 Exhibited Intellectual Characteristics in Intelligence-Enriched Network

After inheriting the eight intelligence abilities of humans, the network will then exhibit following six outstanding characteristics during processing communication events.

(1) Initiative. In terms of thinking for users to relieve them of heavy communication tasks, the network needs to take the initiative to assist them in a respectful way. Instead of waiting for users to report difficulties met or unsolvable problems encountered when performing communication events, the network should forecast the majority of those problems and try to solve them before users experience them.

(2) Considerateness. Users with individual desires and preferences have different requirements on performance for the same type of service. Therefore, it is challenging but necessary for the network to provide customized support to users according to their specific situations in a non-intrusive way.

(3) Responsiveness. The network needs to make their best efforts in immediately responding to any change in user communication status, because users do not expect long delays in the execution of especially real-time service.

(4) Responsibility. The network takes the responsibility to successfully start up a session, to maintain an ongoing session without any perceivable disruption from users' perspective, and to meet performance requirements to the utmost. In other words, the network try not to give up too easily on the connection of a session, nor do they easily drop the session during its execution due to unexpected errors. The network also tries to deliver personalized services according to users' original requirements without easily changing the performance.

(5) Adaptive capability. This ability especially applies to mobile users. Mobile users would like to access the same set of personalized services with familiar consistency as much as possible anywhere, any time, on any usable devices. Therefore, the network needs to be able to

memorize users' customized service requirements, track their communication behaviours, and provide them with proper services wherever they are as they would expect.

(6) Comprehensiveness. In current communications network, all types of equipment, topologies, technologies, and standards coexist and work jointly to meet user needs. This coexistence requires that network designers have a comprehensive knowledge of customer requirements, potential effective communication solutions, possible remedies to manifest problems, and adequate valid measurements. It also requires the designers to be able to embed the knowledge into network design.

In summary, to improve the quality of users' communication life, the network is therefore obligated to accommodate a number of human intellectual competences. Equipped with these competences, the network is able to initiatively participate in users' communications, to enable enhanced personalized services, and to answer users' queries in real time and react to their changes in status. The network is further able to ensure the success of communication sessions, to adhere to users' original preferences regardless of their changing statuses, and to have an overall knowledge of the communication environment.

3.1.4 Network Characteristics Carrying User-centric Competences

To bring intellectual competences into the network, we either create new quantifiable network characteristics or select those related to intelligence from the existing characteristics to carry these abstract competences. These intelligence-related network characteristics are defined in Table 3-1. By validating the new characteristics and evaluating the upgraded ones in later simulation and testing (Chapter 5 to Chapter 7), we can prove the feasibility and correctness of the human-like-intelligence proposal and its corresponding approach (Chapter 4).

Table 3-1. Network characteristics that carry user competences.

Expected Competence	Transformed Network Characteristic	Defining the Network Characteristics That Are Supposed to Carry Intellectual Competence as the Abilities of:	New or Existing
Initiative	Insight	Discerning the hidden or true nature of the network and user situations	New
	Judgment	Weighing up evidence preparatory to making decisions on service delivery	New
	Network intervention	The privilege of investigating session content and taking corrective actions on session connection and execution	New
	Interoperability	Exchanging information between the network and user	Existing
Considerateness	Content awareness	Caring for users through the awareness of the emergency status of their service sessions	Existing
	User-concerned decision-making	Taking into consideration the relationship of session participants in the decision-making process for a session	Existing
Responsiveness	Resource management	Efficiently and effectively deploying all available network resources when needed	Existing
	User profile management	Managing user-profile resources	Existing
	Service life cycle management	Getting sufficient network resources for a service within its lifetime	Existing
	Personal mobility management	Real-time tracking to ensure the successful delivery of network services to users	Existing
	Interoperability	(same as above "Interoperability")	Existing
Responsibility	Quality of service	Guaranteeing a certain level of performance to a data flow and providing different priorities to different users or data flows	Existing
	Security	Improving the likelihood of expectations being met	Existing
	Privacy	Keeping affairs of an individual or a group out of public view and controlling the flow of information	Existing
	Billing	Providing diverse billing manners	Existing
	Content and service protection	Warding off some threats to session execution	Existing
Adaptive capability	Context and service adaptation	Adapting a service's behavior to the changes in context and resources	Existing
	Interoperability	(same as above "Interoperability")	Existing
	User mobility	Tracking a user's location and movements	Existing

	Service mobility	Ensuring the availability of a service on different platforms and devices	Existing
	Session mobility	Suspending and resuming an ongoing session in the network and changing the composition of a service session without risking its safety	Existing
Comprehensiveness	Standardization	Establishing technical standards for intellectual characteristics in the network to bring benefits without causing unnecessary competition	Existing
	Regulatory issues	Forming generally accepted rules for new characteristics with intellectual competences to interoperate with each other	Existing
	Usage viability	Embedding new sessions capable of living and developing under favorable conditions	Existing
Others	Ubiquitous service architecture	Possessing a feasible network architecture with physical components that realize intellectual competences	Existing
	Service platform	Possessing a specific service platform with interface to intellectual network components	Existing
	Middleware for user and service continuity	Possessing proper computer software to connect users, applications, software, and network components	Existing

The ability to fulfill the advanced requirements on the transformed network characteristics will feature the correlation between the network and users. A detailed software design will facilitate realizing such fulfillment (section 3.3.3). Besides, these requirements have set up an evaluation reference for implementing human-like intelligence in the network.

In Table 3-1, most required characteristics have already existed as existing functions in the current network. These characteristics contribute either independently or collaboratively to the implementation of human-like intelligence in the network. (1) However, three crucial attributes – *insight*, *judgment*, and *network intervention* – are still missing in current networks. These three together with the *interoperability* characteristic make up of the *initiative* competence. The *initiative* competence stimulates the network to proactively deploy other types of competence. (2) In addition, the network lacks a mechanism to systemize these competences as members of an intelligence family. If coordinated by such mechanism, these competences will then provide better performance by supplying mutual needs and offsetting mutual lacks.

3.2 Design of Human-like Intelligence Embedded Future Generation Network

Current communications network has been expected to process users' communication events in a user-centric way and to relieve users from the heavy communication burden (section 1.1). We therefore apply human-like intelligence to the network so that the intelligence embedded network can better adapt to the human lifestyle. The adaptation is represented as two network abilities. One is to smartly plan and execute communication sessions according to users' personalized requirements. The other is to independently perform communication sessions with no need for redundant personal efforts from users. We call such intelligence embedded network Human-like Intelligence enhanced Future Generation Network (HIFGN, abbreviated as FGN) and realize them by planting several intelligence components into the existing NGN architecture.

Since we are already clear on the NGN architectural structure, the definition of human-like intelligence, and the importance of infusing intelligence into communications network, the next question will be how we implement human-like intelligence in the network. Suppose that human-like intelligence is implemented as a combination of intelligence components. We first insert these components into the existing NGN architecture and then investigate whether they are compatible with the existing network components (section 3.2.1). After clarifying some important items often used in FGN (section 3.2.2), we focus on four pivotal intellectual mechanisms that work together to infuse the communications network with human-like intelligence (section 3.2.3).

3.2.1 Ubiquitous Network Architecture with Intelligence Component

After having understood the essence of applying human-like intelligence to the communications network, we first need to find out where the original human-like intelligence locates itself and where the intelligence moves to in the network-user picture. Figure 3-2 exemplifies how NGN imports human-like intelligence in the NGN-user architecture.

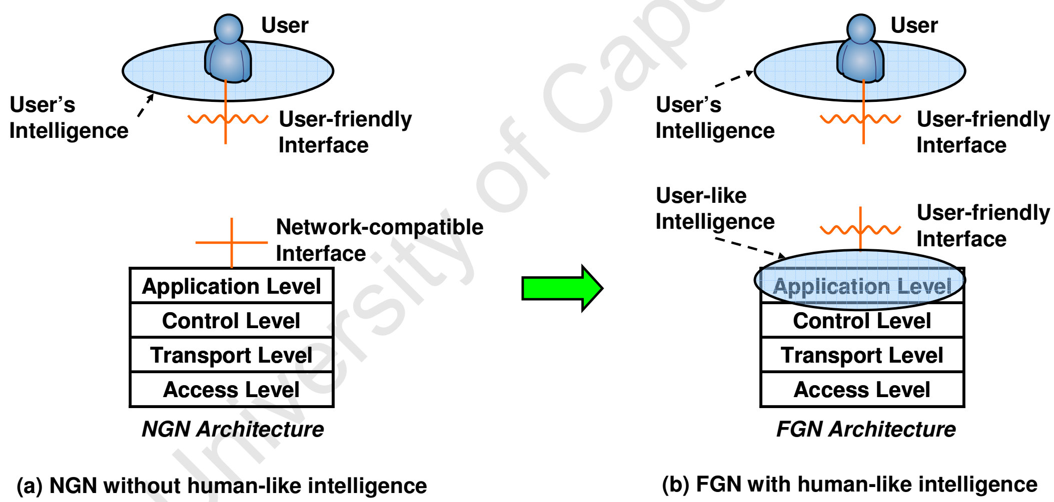


Figure 3-2. Importing human-like intelligence into NGN architecture to form FGN architecture.

The sequential figures from Figure 3-2 (a) to Figure 3-2 (b) have exhibited a successful drift of human-like intelligence from a user to NGN. The user who initially possesses full human-like intelligence talks to NGN via a user-friendly interface. NGN has its top level – the application level – communicates with the user via a network-compatible interface.

In Figure 3-2 (a), the user with full intelligence may find that NGN without human-like intelligence is unable to assist with the communication process as expected, because NGN cannot think in his/her shoes. The user may also consider NGN unfriendly because NGN does not serve him/her with a user-friendly interface but a network-compatible interface. That is, the user and NGN talk in different languages and they therefore cannot communicate properly.

Suppose user intelligence can clone itself and the cloned part can drift from the user to NGN. The original and cloned parts of user intelligence communicate with each other using user-friendly interface. After the cloned intelligence has reached and converged with NGN as shown in Figure 3-2 (b), the human-like intelligence embedded NGN (i.e., FGN) becomes more capable of providing the user with personalized services. With the intelligence, FGN is able to use its initiative to take the user's desires and requirements into consideration when processing communication events. It is also capable of friendly talking with the user in a mutually agreed language, using the user-friendly interface that can be easily implemented through the cloned intelligence. Moreover, the user-friendly interface makes it easy for the user to modify his/her communication profiles in FGN. In addition, the cloned intelligence is able to take over some of the tasks that the user had to perform in the past by participating in and facilitating his/her busy communications. In this way, FGN is able to relieve the user from heavy communication tasks.

3.2.2 Clarification of Pivotal Elements and Parameters in Design

Before describing the main mechanisms of "intelligentizing" the current communications network, we first need to clarify some important elements and parameters that will either carry human-like intelligence or contribute to the realization of human-like intelligence in network design.

(1) Future generation network (referred to as "intelligence network"). The intelligence network refers to the heterogeneous network that has nearly the same structure as that of NGN but enhanced with one additional feature – human-like intelligence. For differentiation, we call the current communications network without human-like intelligence a "general network".

(2) Network user (referred to as "user"). A user in the intelligence network can be any social entity in reality, including a private person, a group of people, a terminal, an IP address, a technology, a network, or software.

(3) Network user's human-like intelligence (referred to as "user intelligence"). User intelligence is human-like intelligence defined in section 3.1.2.

(4) Human-like-intelligence part in the intelligence network (referred to as "human-intelligence part"). We develop mechanisms to intelligently mimic network users' communication behaviors and attitudes. The intelligence network then separates an area of hardware resource to store these mechanisms in the form of software. The hardware and software parts within the intelligence network are together called the human-intelligence part. The rest of network resources in the intelligence network are called "physical-network part".

(5) Network service (referred to as "service"). In general, a service in the communication world means a process that creates benefits by facilitating a positive change in network users' communication life. In a narrow sense, a service means a function that the network offers to meet the communication requirements of a user.

(6) Service session (or "communication session", referred to as "session"). A service session is a lasting connection between a user that requires a communication service and a provider that offers the service. It can also be a connection between a user who initiates a service and the peer that receives the service. A session implements an instance of the communications network delivering a service to network users.

(7) Application. If a service means the function that needs realizing between a pair of users, an application for the service means a physical realization of the service on specific devices and links. That is, a user delivers a communication service to another user via an application. For example, the ability of the network to provide video streaming to network users is a service. The service's realizations can be streaming video on Internet, on-demand video on a broadband television, or low-capacity video chips on a third-generation (3G[37] [78]) cell phone.

[37] 3G is the third generation of wireless-communication standards and technologies that import high-speed internet access and video telephony into the existing second-generation mobile systems.

(8) Human resources. These resources include network users, their communication devices, and their participation in communications.

(9) Physical network resources (referred to as “network resources”). These resources include network equipment, technologies, and network standards.

(10) Information resources. Information resources are the communication contents being carried by network sessions. Both network resources and human resources materially exist, whereas information resources come into being with the birth of network sessions and vanish when these sessions are over.

(11) Communication resources. In the intelligence network, communication resources comprise all types of resources, including network-, human-, and information-resources.

The intelligence of the network shows up when these network elements and parameters dynamically act on and affect each other.

3.2.3 Major Mechanisms Needed for Implementing Intelligence in Network

With human-like intelligence embedded, the network is able to sequentially perform a series of human intellectual actions in an ongoing communication event. These sequential actions include (1) detecting problems in an event if they exist, (2) analyzing and concluding the cause of these problems, (3) suggesting reference solutions, and finally (4) making proper decisions on how to appropriately process the event. The following four sections will illustrate in detail how to embed these four types of human ability in the intelligence network.

3.2.3.1 Detecting Problems in a Communication Session

Since we intend to relieve users from the heavy communication burden (section 1.2.2), the concerns are what problems may cause users an extra burden during their communication and what technical issues are responsible for these problems (section 1.2.1). (1) Lack of a universally agreed user-centric service platform under heterogeneous networks may hamper the mobility of a session, a service, or a person. For example, currently most communication sessions occur within the same network because no proper service platform is able to seamlessly route a session or a service from one type of network to another. (2) Lacking the ability to detect the uncertainties caused by humans, users may fail a communication event at anytime, especially at the connection stage. For example, a call session initiated to the user Lisa may not finally get to her because she is busy on another call. (3) Current communications network has not given its best efforts to delivering services to users and has therefore created difficulties for users to easily perform communication tasks. For example, nowadays, Lisa has to remember all Bill's contact details such as office phone number, email address, and Yahoo Identification (ID) ([79]) if she wants to contact him in different ways. The network is unable to provide Lisa with a unique contact entry that relates to Bill's all three physical entries. (4) Network-, human-, and information-resources are wasted due to inefficient and improper use of individual resource types and a lack of interaction between them. For example, when Lisa is on a call, she cannot answer a second call. Lisa's officemate Bill is free but no calls are directed to him. Bill (human resource), Bill's office phone (network resource), and the content of the second call (information resource) are all wasted. (5) Users' right to use communication resources is seriously impinged when the current communications network sometimes sacrifice the users' first desire to solve the technical problems encountered, such as the NS/EP problem (section 1.2.1). (6) The current network has imposed repetitive work on users by looking at each one as an individual communicating party. For example, twin sisters each get a cell phone as birthday gift. They go through the same steps to set up their respective profiles on their own cell phones. The network cannot offer them the service that one sets up her profile on her cell phone and the other copies her sister's profile and only makes minor changes related to her social relations. Hereof both the submitter and beneficiary of the shared profile are individual users. The "existing settings" service provided by Vodafone also allows a user to select a profile template and make modifications accordingly [80]. Yet the service is not personalized enough as required because its profile submitter is the network and not users. (7) Little research is concerned with the unavailability of communicating parties and the unreasonable human-resources distribution over different access networks. (8) Lacking the ability to provide users with as many personalized services as possible adds to user antipathy to use these services.

Because so many problems can arise from users' communication status and subsequently affect their communication manners, it is necessary for the intelligence network to be able to detect and remedy these problems. One suggestion is to intentionally interrupt an ongoing communication session for a reasonably short period for problem identification. By interrogating the temporarily interrupted session what problems it may have met, the intelligence network is able to detect existing network problems and diagnose them according to the answers of the session. Figure 3-3 shows such a mechanism in the NGN architecture:

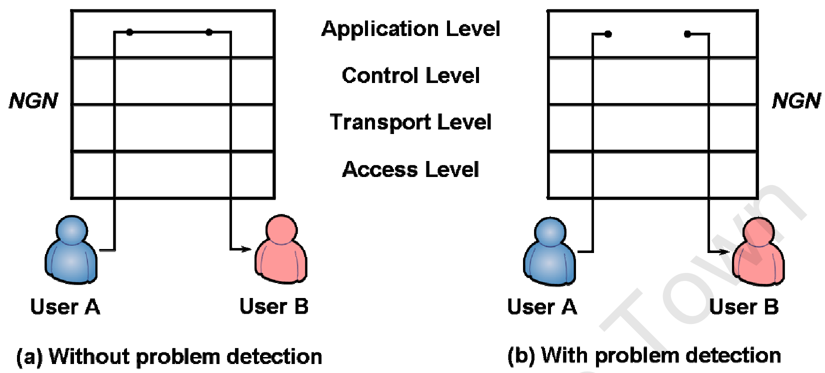


Figure 3-3. Problem-detecting mechanism.

User A initiates a service session to User B in a general NGN architecture. If NGN has not encountered any abnormal situations, the session will go from User A, through the access-, transport-, control-, and application-levels to compose the application. Finally, the application will be carried by the session to User B as shown in Figure 3-3 (a).

However, the intelligence network needs to take the initiative to enquire into a session about the problems that it might have encountered. We thus break the service session at the application level and keep the session in the intelligence network for a fixed period as shown in Figure 3-3 (b). The break leaves the intelligence network enough time to detect the potential problems and come up with optimal remedies (sections 3.2.3.2 to 3.2.3.4).

3.2.3.2 Analyzing Problems Using Human-like Intelligence

On detecting the problems in a session, the intelligence network needs to immediately identify the nature of the problems and the causes for them. The application level of NGN does not perform a problem diagnosis for the session. One remedy is to ask the proposed human-like intelligence to help with the diagnosis as shown in Figure 3-4:

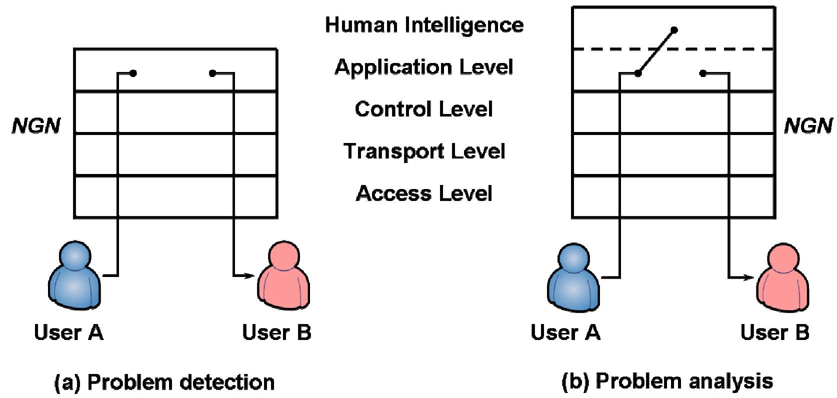


Figure 3-4. Problem-analyzing mechanism.

Figure 3-4 exhibits two scenarios where the intelligence network has just detected the problem as in (a) and where it asks the proposed human-like intelligence to diagnose the problem as in (b). Because the actions of recognizing and diagnosing user-concerned problems are performed in the human-intelligence part, they will not cause extra work for the existing NGN.

3.2.3.3 Proposing Solutions to Improve Communication Efficiency

After having revealed the user-centric problems in the network (section 3.2.3.1), the human-intelligence part comes up with optimal practical steps to control and finally solve these problems. The basic objective for proposing optimal solutions is to improve communication efficiency. An efficient network is able to ensure that a single service session is as successful as possible, to drive a larger number of successful communication events in a fixed period, to utilize the network resources to the utmost, and to make optimal use of human resources. Figure 3-5 illustrates the mechanism of human-like intelligence proposing solutions.

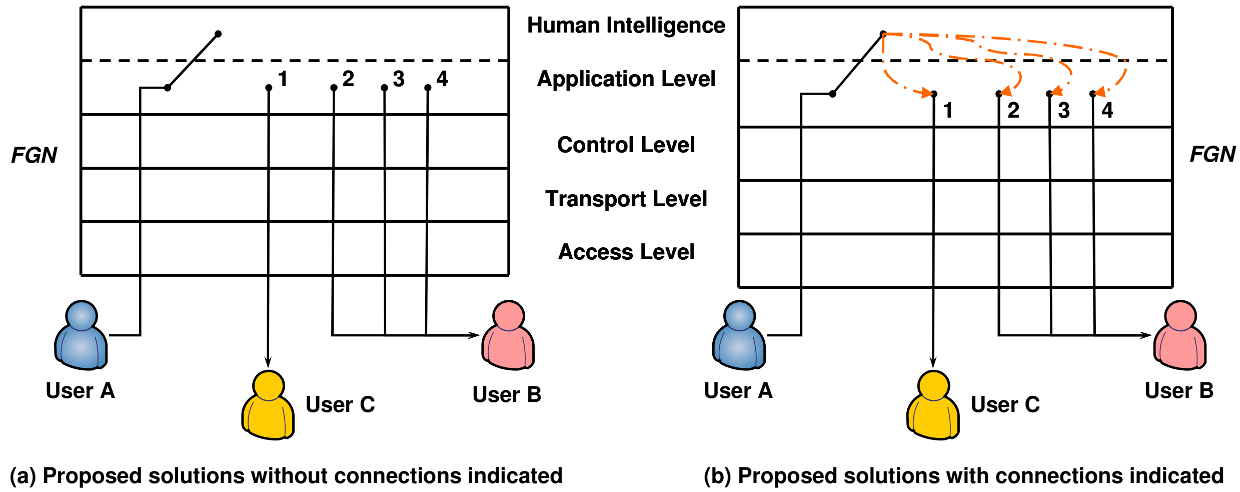


Figure 3-5. Solution-proposing mechanism.

Figure 3-5 exemplifies a scenario where the intelligence network has proposed four manners for User A to deliver a certain service. One manner is to deliver the service to the third user – User C, and the rest to any of the three available devices of User B.

Using the solution-proposing mechanism, the human-intelligence part looks up all the possible connecting manners[38] from User A to some resource in service lifetime. With so many options to choose from, the intelligence network is able to deliver the session as successfully as possible. What is worth mentioning here is that the mechanism attaches individual benefits to each proposed connecting manner so that future mechanisms are able to make proper judgment for each manner. Some available connecting manners are able to meet User A's original needs, for example, manner 4 to User B. Some make better use of available network resources, such as manners 2 and 3 to User B when manner 4 is occupied by other sessions. Some manners try to use other free human resources when the originally expected human resource is unavailable, such as manner 1.

[38] Connecting manner is a set of physical devices and links for session execution.

3.2.3.4 Making Decisions based on Available Communication Resources

Besides analyzing the nature of occurring problems and proposing solutions to solve them, the human-intelligence part makes decisions on the final service-delivery manner.

Because the focus is to make the best use of communication resources while sparing human labor, the human-intelligence part decides the final delivery manner by fully considering network users' communication issues. These issues include the session initiator's original requirements on the session, available network resources for the expected receiver, personal availability of the receiver, availability of other potential service involvers, and the social relations of the initiator, the receiver, and other involvers. The essential intelligence shows up when the transformed intelligence network makes decisions on how to rationally deliver a service session according to user communication issues. Figure 3-6 exemplifies a scenario where the decision has been made based on the scenario described above (section 3.2.3.3).

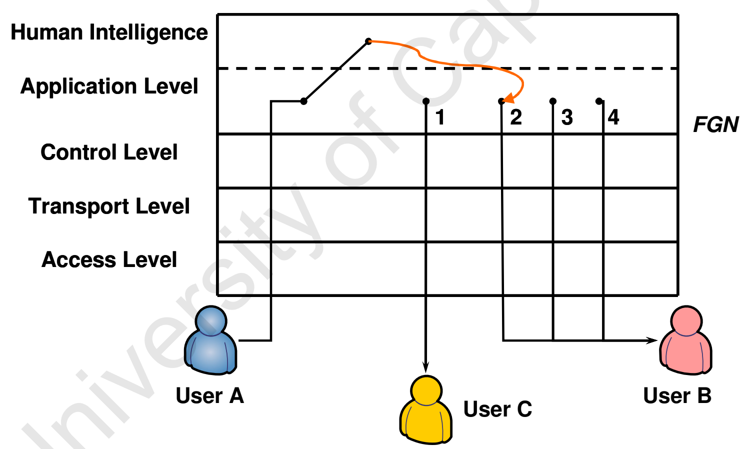


Figure 3-6. Decision-making mechanism.

In Figure 3-6, among the four available connecting manners, the human-intelligence part finally chooses manner 2 to deliver the service from User A to a device of User B. This decision is made on condition that User B is free, the device for manner 4 is busy, the cost on manner 3 is more expensive than that on manner 2, and User A does not trust User C in receiving the service.

3.3 An Approach to Future Generation Network

We propose a virtual-user approach to realize the vision of planting human-like intelligence into the communications network. This section will briefly exhibit the concept of virtual user (section 3.3.1) and set up a virtual-user model to present how virtual users interact between themselves and with their presented real users (section 3.3.2). Thereafter, the section will point out the major features that the model holds (section 3.3.3).

3.3.1 Establishing a Virtual User to Represent Human-like Intelligence

A virtual user is defined as the representative of a real user in the intelligence network. The relation of a virtual user to a real network user is similar to that of human-like intelligence to human beings. That is, a virtual user has the logical analysis and thinking abilities of a real user in conducting communication events. It is able to discover the problems that the real user may come across during his/her communication process, analyze and solve these problems, and resultantly make proper contributions to the communications as the real user would. One virtual user only works for a specific real user.

In the real world, each network user possesses a specific communication profile that comprises three parts: personal details, current communication status, and social relations with other users. When involved in a communication event, the network user refers to parts of his/her communication profile to get the event successfully executed. Thus the virtual user, which is supposed to conduct communication on behalf of the real user under certain circumstances, needs to make a thorough, real-time reflection of the real user's communication profile.

Many existing network policies and technologies have been focusing on mapping real users' personal details such as their names and on tracking the users' present communication status such as their presence statuses. However, few of them address the issue of using communication-event involvers' social relationships to assist in monitoring the event. In fact, the event involvers' social relationships strongly facilitate the execution of the communication event by availing the event of more network resources and spare human resources, balancing network traffic, and allowing session and personal mobility. Devoting attention to the effect of network users' social relationships on their communications is necessary and worthwhile [81].

If the virtual-user design succeeds, virtual users are able to partake in real users' heavy communication tasks and easily extend the function of the intelligence network to real users. (1) Because the virtual user keeps the real user's communication profile, it has a better chance of providing the real user with a friendly interface, with which the real user can easily make his/her own preferred services and leave out unnecessary communication events. (2) For the same reason, the virtual user is very informative for the intelligence network. If there is a need to refer to the real user for indications on the session process, the intelligence network can simply go to the virtual user instead of the real user, thereby saving time and energy for the network.

3.3.2 Architecture of Virtual-user System

The architecture of the virtual-user system is in fact very simple. In the system, each real user has a virtual-user representative in the intelligence network. The virtual-user representative keeps a copy of the real user's personal details, real-time tracks of the real user's present communication status, and relates to other virtual users via the social relationships of the represented real user. Because the virtual user is obligated with so many responsibilities, we will therefore draft a detailed function design of the virtual-user system (Chapter 4), and prove the design by implementing it in a software environment (Chapter 5 and Chapter 6) and evaluating its performance (Chapter 7). Here we first set up two communication scenarios to help in understanding the functional role that the virtual user plays in communication events.

Before exploring the two scenarios, we introduce the diagram language as shown in Figure 3-7 to illustrate the manners describing the figures in the following context.

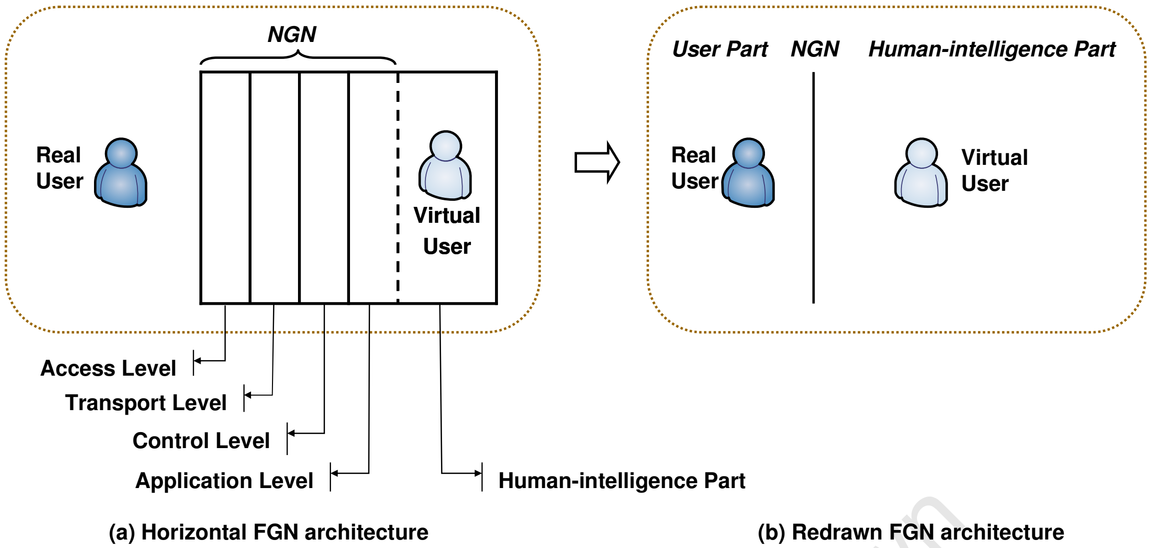


Figure 3-7. Illustration of a diagram language.

For an easier view and realization of the intelligence network, we transpose the FGN architecture (NGN architecture with human-like intelligence embedded) from a vertical orientation to a horizontal orientation as shown Figure 3-7 (a). Because the following figures in this section will focus on the procedure that the human-intelligence part assists users' communication, we redrew Figure 3-7 (a) as Figure 3-7 (b) where more emphasis is given to the human-intelligence part in the transformed FGN architecture. We compress the NGN architecture with four levels into a vertical line because the structure of NGN has nothing to do with human-like intelligence.

3.3.2.1 Communication Scenario I: between Two Users

A service session manipulates a service-process flow that starts from a session initiator, goes through the network, and ends at a session receiver. We use a normal phone call to represent a service session and then call the session initiator "caller" and the session receiver "callee". The service that the session carries is to complete the phone call. A third party who has also contributed to the session is called "assistant callee". The caller, callee, and assistant callee's virtual representatives are respectively called "virtual caller", "virtual callee", and "virtual assistant callee". Using this phone call as an example, Figure 3-8 compares the difference between the session processes with and without human-like intelligence's participation.

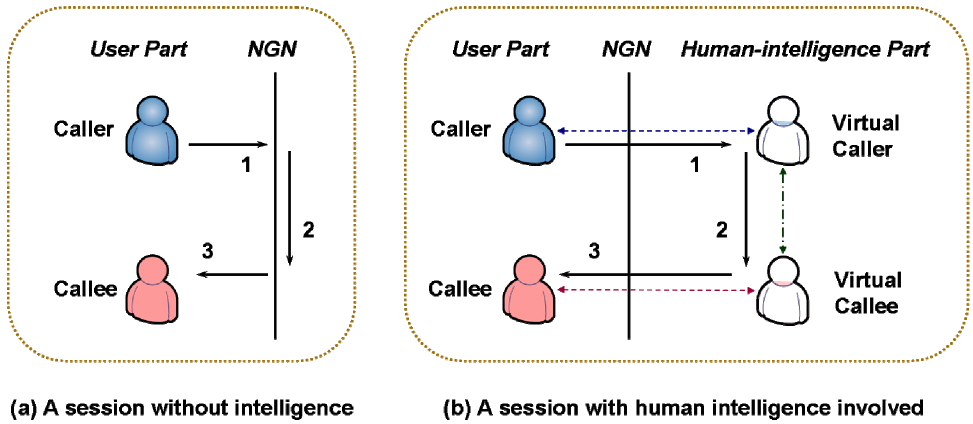


Figure 3-8. A two-party communication scenario.

The dashed bi-directional arrowed lines indicate the physical and functional connections between the real user and its virtual user. The long-dashed-dot bi-directional lines between any two virtual users imply a social relation between the two real users. The sequence of the omni-directional arrows demonstrates the procedure of a service session.

Figure 3-8 (a) depicts the communication scenario in NGN and Figure 3-8 (b) visualizes the communication scenario in FGN.

In current communications network, as shown in Figure 3-8 (a), a call starts from the caller (step 1), traverses over NGN (step 2), and finally gets to the callee (step 3). If the call can finally reach the callee, we say that the call service has been successfully delivered. Whereas in the NGN architecture, if either the physical network is not working properly or the callee is unavailable, the call will fail and the call service cannot be delivered to the callee.

The process of a call session in the intelligence network is more complex in Figure 3-8 (b). When the caller starts a call, the call session penetrates the physical NGN architecture and gets to the virtual caller (step 1) in the human-intelligent part. The virtual caller negotiates with the virtual callee on whether the callee is capable of receiving the call session at that moment (step 2). If the virtual callee finds that the callee is technically available (being with proper and available device at the appropriate moment) and is personally free (not being busy with other social affairs), it will direct the call to the callee. In this way, the intelligence network establishes a successful call session between the caller and the callee with the assistance of virtual users. Figure 3-9 further illustrates how virtual users determine an optimal session-delivery manner.

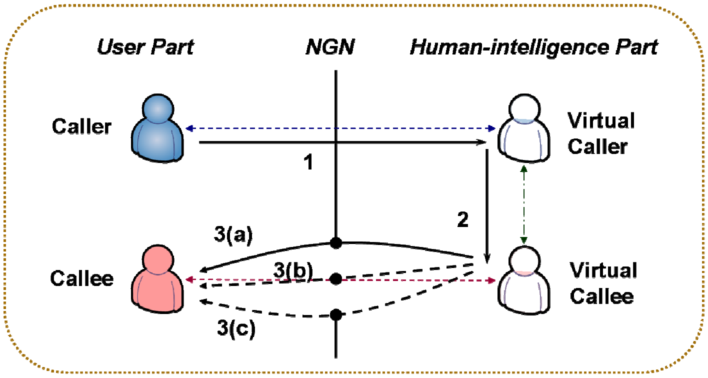


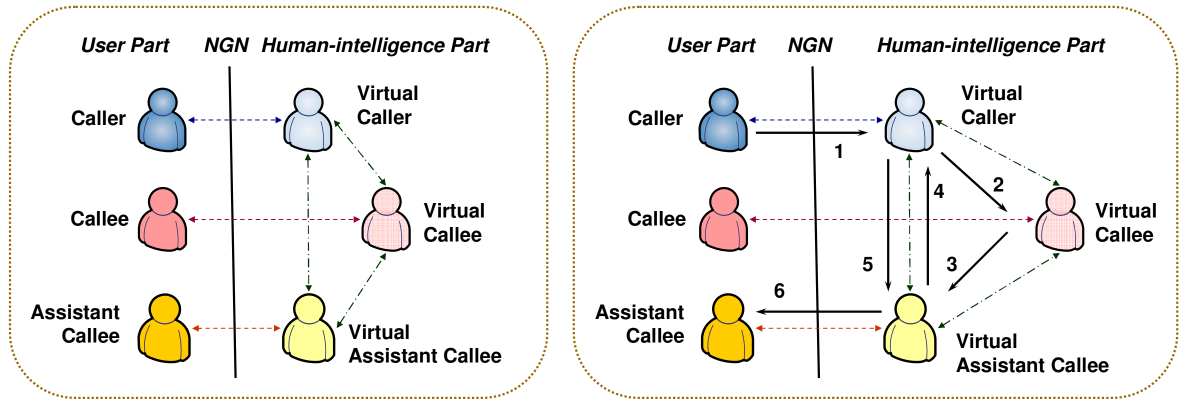
Figure 3-9. A two-party scenario with three optional delivery manners.

Figure 3-9 exhibits a scenario where the virtual callee has found three optional manners to deliver the call service. Suppose the connecting manner 3(a) is the desired one by the caller. (1) If the virtual callee finds that the manner 3(a) is unavailable whereas other two are, it will use any of the two available manners to deliver the session, thereby increasing the success probability of the session. (2) If the virtual callee finds that the manner 3(a) provides a worse performance than the other two such as poorer quality or more expensive, it will suggest that the network use two other connecting manners, thereby improving service performance.

However, if the virtual callee finds no communication manner available for the callee, what will the networks do – fail the session or think of other ways?

3.3.2.2 Communication Scenario II: among Three Users

In the case where the virtual callee does not find any communicating manner available for the callee in Figure 3-9, the intelligence network will come up with a fourth option to deliver the call session. It will involve a trustworthy third party in the session to help with the session delivery. Figure 3-10 visualizes such a scenario where the virtual-user system has identified a third party – assistant callee – with the assistance of human-like intelligence.



(a) Mappings and relations among users and virtual users **(b) Processing a session with the help of human intelligence**

Figure 3-10. A three-party scenario – requesting third user's participation.

The three-party communication scenario in Figure 3-10 is a continuity of the two-party scenario in Figure 3-8 when the virtual callee finds that the callee is completely unavailable for the call session. In Figure 3-10, when the virtual callee finds that the callee is not available in terms of the proper communication manner and the proper answering time, it will search its socially related parties to establish whether other people are personally available and technically able to receive the call. After investigating a particular virtual assistant callee, the virtual callee finds that its represented real user can handle the service. The virtual callee then authorizes the virtual assistant callee to handle the service after several rounds of negotiation (step 3) and at the same time it informs the virtual caller of the situation. The virtual assistant callee volunteers to the virtual caller to continue the session (step 4). If the virtual caller trusts the virtual callee and the virtual assistant callee, it will agree to redirect the session to the assistant callee via the virtual assistant callee (step 5). After all these appeals, choices, and acknowledgements between the virtual users, the networks finally select the assistant callee to receive the call session (step 6).

Yet if the callee has never set any criteria of dealing with this type of service, the virtual callee will find that it has no idea of how to perform the session. In this scenario, the virtual callee may also turn to other virtual users. Once having found a trustable assistant callee who knows how to process the call service, the virtual callee will learn from the virtual assistant callee on what to do with the session and finally deliver the service using any manner provided in the above two scenarios. A sample of turning to the assistant callee for instructions on service delivery and finally delivering the session to the callee is shown in Figure 3-11:

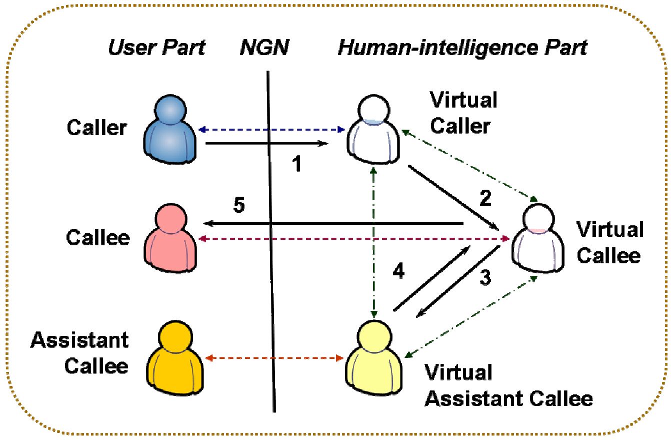


Figure 3-11. A three-party scenario – asking third user for session-delivery method.

Figure 3-11 illustrates a scenario where the virtual callee learns from the virtual assistant callee the methods of service delivery. After learning (steps 3 and 4), the virtual callee is able to suggest that the intelligence network successfully delivers the call to the callee. Besides, once a virtual user has accomplished a successful session by choosing the proper assistant user, it files in its memory the way to process this type of service under certain conditions. The virtual user thus has learnt the methods of successfully performing the service without the real user's efforts. Therefore, if the same type of service is to reach the user again, the intelligence network is able to deliver the session using the latest optimal connecting manner stored in its memory.

3.3.2.3 Summary of Benefits from the above Communication Scenarios

The communication scenarios depicted in Figure 3-8 to Figure 3-11 respectively illustrate the steps by which the virtual-user system investigates optimal session delivery manners. (1) Once a session successfully reaches the human-intelligence part, the session automatically breaks for indications on optimal delivery manners. This breaking action is proactive and it gives the network a chance to establish the presented problems. (2) The virtual callee then makes the effort to enquire as to the status of the callee and establishes all the possible manners of call-service delivery. Via the problem-diagnosing and solution-proposing processes, the network is able to properly react to the identified network problems at the most appropriate moment. (3) After consulting some higher authority and obtaining the permission, the network finally delivers the service using the most appropriate connecting manner among the available ones. This procedure involves the ability to make an intelligent decision.

By going through such an intelligence procedure, the virtual-user system is beneficial in terms of increasing the success rate of session delivery, improving service performance, making the best of network- and human-resources, sparing human labour, and providing session mobility through the extensive use of communication resources. (1) The success rate of the session increases when the virtual users make an effort to seek alternative routes to deliver the service if the original connecting manner is unusable. (2) The network may also improve service performance when the virtual users find better solutions than the original solution, such as a cheaper way of delivery or a more robust link to the callee. (3) In all these scenarios, only two real people with proper devices are actually involved in the call session. They are the caller and the final callee – either the expected or assistant callee. The unavailable or less-qualified network resources (i.e., manner 3(a) in Figure 3-9) and the unavailable human resources (i.e., the busy callee in Figure 3-10) are not accommodated into the final delivery at all. (4) Furthermore, human labor is saved because these two actually involved users are only respectively responsible for initiating a service session and receiving a successful incoming session. The human-intelligence part takes care of the complex part of the work such as suggesting better connecting manners, looking for a trustworthy assistant, and setting up an optimal connection between the two most appropriate users for the session. (5) In addition, the session, the service, and the user are more flexible in mobility when the network uses human-like intelligence to carry them out.

3.3.3 Model Featuring the Cooperation between Networks and Users

After having sketched a structure for the virtual-user system and having illustrated the system with the above communication scenarios, we are clearer on the benefits of setting the virtual user to realize human-like intelligence in the network. The virtual-user approach helps the network to better interwork with real users, to increase session-delivery success rate, to improve service performance, to make the best use of available network- and human-resources, to save human labour, and to provide session-, service-, and personal-mobility.

Because the virtual users are capable of so many functions, they are able to fulfill the requirements for the intelligence that a network needs, including initiative, considerateness, responsiveness, responsibility, adaptive capability, and comprehensiveness (section 3.1.3).

Firstly, virtual users take the initiative to enquire from the expected session receiver of his/her current communication status and to suggest optimal delivery manners for the session. The network carries on these actions without causing extra work to real users and is therefore able to relieve real users from communication overload while maintaining a high-quality service.

Secondly, each virtual user keeps a frequently updated communication profile of a specific network user. Therefore, the virtual user knows the exact communication tastes of the network user and is able to serve the network user in their shoes.

Thirdly, the requirement on responsiveness is met in the virtual-user system when every session stops and waits for an indication from virtual users. With the stop-and-wait mechanism, the intelligence network responds to all the problems in each ongoing communication session.

Fourthly, virtual users take full responsibility for ensuring a communication session to be as successful as possible by observing and analyzing the communication situation and suggesting the most suitable communicating manners for the session. Strong communicating capabilities refer to the abilities of delivering a session with the highest probability of success, delivering a session with the best performance, and meeting real users' requirements to the utmost.

Fifthly, the smart choice of the most capable communication manner for the expected session receiver and the involvement of new network users in a suspended session endow the network with adaptive capabilities to provide as many personalized services as possible.

Lastly, we demonstrate our comprehensive understanding of the human-intelligence concept by implementing one of its approaches – the virtual-user system – in a software environment. Whether the system works or not and whether it is able to provide the desired intellectual abilities will be proven by the experimental results in the upcoming chapters.

3.4 Prospective Life-case Application of Human-like Intelligence

Network users' communication life will benefit from the intellectual process in the human-intelligence enriched network. The example below narrates four life-cases where the intelligence network individually deals with four types of communication services by making the most appropriate decision. Each life-case requires the network to go through the intellectual process of detecting a problem, analyzing the problem, proposing solutions, and making decisions on which solution to use. The relationships of the people in the example are shown in Figure 3-12, where Mr. Smith is the central person of all communication sessions and others are involved in different service sessions. These people all use the intelligence network to manage their communications.

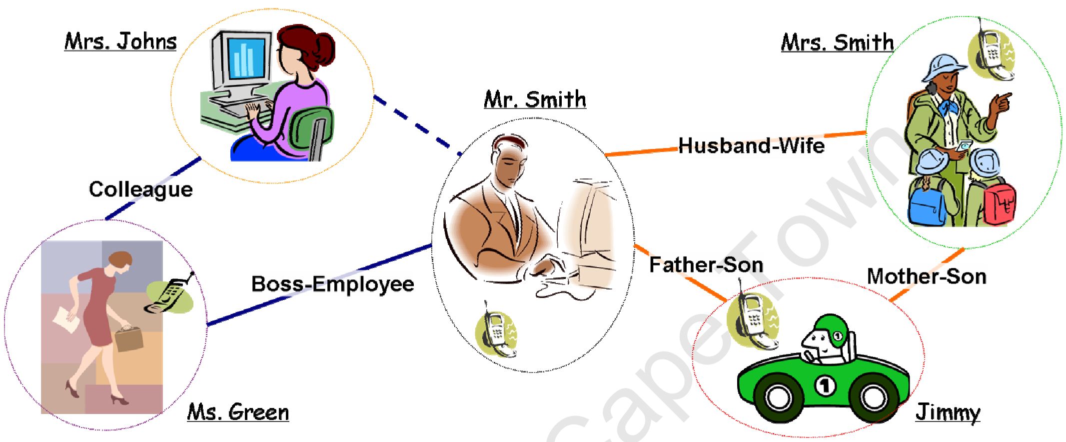


Figure 3-12. Mr. Smith's communication life.

Figure 3-12 illustrates Mr. Smith's social relationships and current communication status. Mr. Smith is a product manager. He is currently in his office and is therefore able to access Internet and his cell phone. Mrs. Smith, who is a teacher, goes for an outing with the kids and is available on her cell phone. Their son Jimmy carries his 3G cell phone that has a big screen to view pictures. Ms. Green – Mr. Smith's present secretary – is busy with an exhibition and is thus only accessible via her cell phone. Mr. Smith's former secretary – Mrs. Johns – is currently in her office and is available on Internet.

Suppose a service session reaches Mr. Smith in the communication environment described above. After going over the intellectual ways of processing the service, we will be clear about how the intelligence network succeeds in providing the users with customized services by making the most appropriate decisions.

Today is Jimmy's birthday. Mr. Smith sends to Jimmy's computer a birthday eCard[39] and he wishes Jimmy could read the card as soon as possible. The virtual Jimmy finds that Jimmy is only available on his 3G cell phone and thus decides to send the picture to his 3G cell phone. By carrying out such a decision, FGN keeps the original service function of delivering an eCard, although it has to sacrifice the desired device.

Meanwhile, an insurance company contacts the Smith family to take a survey of the family health. The survey is supposed to go to Mr. Smith because he has done this before. Yet, Mrs. Smith is also eligible for the survey as an adult family member. The call first gets to Mr. Smith's cell phone but at the time Mr. Smith is busy sending Jimmy the eCard and he does pick up the call. FGN then forwards the call to Mrs. Smith's cell phone. If Mrs. Smith answers the call, the survey can be done. This case shows that the successful delivery of a service session relies on the communication availability, the social relationships, and the learning ability of the session involvers.

A few minutes later, Mr. Smith sends an Email to Ms. Green, asking her to organize a meeting next week. As the intelligence network finds that Ms. Green is offline on Internet but available on her cell phone, it will send a short message to Ms. Green's cell phone about the content of the Email. That is, the network finds an alternative way to deliver a service to Ms. Green when the desired one is unavailable, without changing the communication content and causing any inconvenience to Ms. Green.

Supposing that the virtual Ms. Green does not know the trick of converting the Email to short message, the intelligence network will not easily fail the service. After having found that the experienced Mrs. Johns is available online, the network suggests that the virtual Ms. Green learns from the virtual Mrs. Johns on how to deal with such service. Once having learnt the process relevant to the service, the virtual Ms. Green then performs the same actions as described above as well as memorizes the methods of dealing with the communication events.

---

[39] eCard is a the production of a technology that provides a greeting-card service using digital media.

3.5 Summary of Chapter 3

This chapter narrated the proposal of applying human-like intelligence to the current communications network. After a careful analysis of the current network' needs for accommodating human nature, we finalized the required human characteristics and identified the network characteristics that are able to carry these natures. Embedded with these human characteristics, the network is supposed to be able to detect network problems, analyze these problems, propose rational solutions, and choose the most appropriate delivery manner for a service session. Having proposed human-like intelligence to embody these abilities of the network, we further modelled human-like intelligence as virtual users to intelligently serve network users in service sessions. At the end, we gave a life-example to illustrate how human-like intelligence functions in the intelligence network to facilitate network users' communication life.

Chapter 4 will depict the detailed functional design of the proposed virtual-user system, including the system architecture, the basic elements, the function modules, the major mechanisms, the key parameters, the process flows, and the potential implementations.

Chapter 4 Designing a Virtual-user System to Intelligently Perform Communications

To better serve network users, we abstract human-like intelligence from humans and apply the intelligence to the current communications network (section 3.2), thus expecting more benefits for users from the human-like-intelligence enhanced network.

We establish a virtual-user system to simulate a community of virtual users in the network (section 3.3). The system is equipped with the mechanisms of detecting problems, analyzing problems, proposing solutions, and making decisions for communication sessions. In theory, the virtual-user system is likely to be a practical and effective approach of embodying human-like intelligence in the network because the proposed virtual users are strongly capable of mimicking human nature. These imitations are with respect to the properties of initiative, considerateness, responsiveness, responsibility, adaptive capability, and comprehensiveness (section 3.3.3). This chapter will strengthen this theory from a technical point of view. Figure 4-1 first depicts the functional design of the virtual-user system.

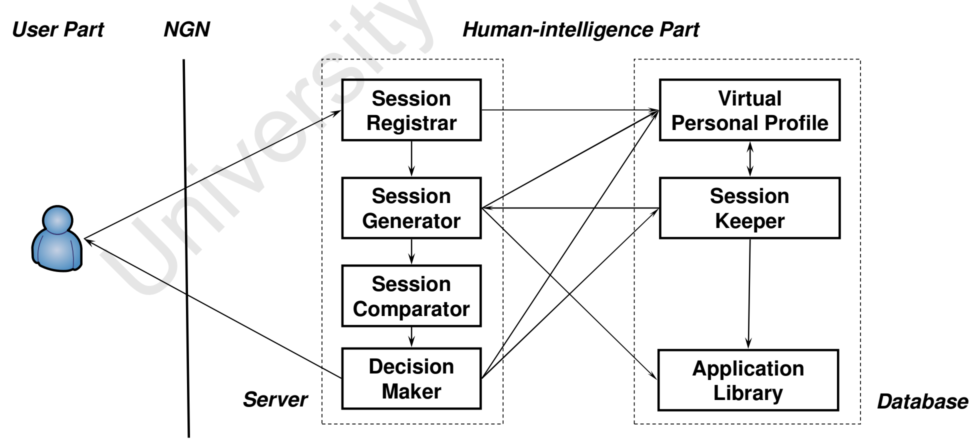


Figure 4-1. Functional design of the virtual-user system.

In Figure 4-1, seven function modules make up a virtual-user system: Session Registrar, Session Generator, Session Comparator, Decision Maker, Session Keeper, Virtual Personal Profile, and Application Library. On receiving an ongoing session from a network user via the current communications network, the system calls on all the function modules to work out an optimal session-delivery manner. It then instructs the physical network to use the resulting manner to deliver the session to another user.

The virtual-user system possesses several unique features. Firstly, the system only manipulates virtual sessions. A virtual session images a real communication session in the virtual-user system by a course of software programs. It possesses the major features and essential functions of the real session. The system does not participate in a real-session execution in the physical network and only works as a consultant to propose an optimal delivery solution for the execution. Secondly, the system manages a group of socially related virtual users as well as conserves the independence of each virtual user. Thirdly, the system performs a large number of calculations and operations over databases to carry out quantified human-like intelligence.

In the following sections, we introduce basic system elements (section 4.1), system modules with each manipulating an independent function (section 4.2), major mechanisms that work in-between these modules (section 4.3), and the workflow of the system in six scenarios (section 4.4). We then describe the key parameters for quantifying and evaluating the functions (section 4.5) and narrate two potential implementations to exhibit its practicability (section 4.5). In what follows, we use FGN and NGN to respectively represent the intelligence network and the current communications network (i.e., the physical-network part of FGN).

4.1 Basic System Elements

The virtual-user system manipulates three types of system element: user-, application-, and session-elements. These elements themselves carry service information and, by flowing from one function module to another, transport the information between the modules. To embody advanced intelligence features and convey meaning information to the modules, these system elements need to possess specific characteristics and perform complex tasks.

4.1.1 User Element

A network user in FGN comprises two parts. One part is the real user. The other part is the virtual user, namely, “virtual personal profile”. We uniquely define a user in the physical network by storing his/her communication information in the virtual personal profile.

A user element is a once-off software realization of such a virtual personal profile in a specific virtual session. It contains the real user's full information in terms of personal details, current communication status, and social relations. (1) Personal details include user ID, user name, social trustworthiness, emergency status, default activity locations, and default available communicating devices at each location. (2) Current communication status comprises present social activities, locations where these activities occur, and present available devices at each location. (3) Social relations are a list of other network users who have certain social relationships with the principal user and present their reliability of communicating with the user using a set of trust-related issues. Each social relation is described as a relation type, relation domain, and demanding degree. Hereof and in what follows, a principal user is the core user whom we need to analyze when establishing social relationship information.

User elements possess both static and dynamic features. (1) A single user element stores a user's static communication information in fixed data fields. The information includes personal details, communication status at a specific moment, and social relations. (2) A series of user elements for the same user are able to represent the dynamic communication status of the user. User communication status generally changes with time. The human-intelligence part creates different user elements for a user at respective moments when these elements are requested. A user element created at a specific moment is able to represent the user's status at that moment. Therefore, a systematic combination of a group of such elements is able to represent the changing communication status of the user. That is, user elements are time-sensitive.

By referring to the user information conveyed by these user elements, the system modules are able to carry out their individual contribution to realizing human-like intelligence. Virtual Personal Profile provides the sources for generating user elements by reading user information from the user database and storing it for further use. Session Registrar updates a user's communication information in Virtual Personal Profile by sending it the user elements that contain the user's most recent communication information. Service Generator is only able to generate a virtual service session after having obtained the user elements of all session involvers from Virtual Personal Profile. Session Keeper frequently inquires of Virtual Personal Profile for users' most recent information by asking for relevant user elements and, when necessary, gets updated by Virtual Personal Profile through the user information carried in user elements.

4.1.2 Application Element

A communication service stands for a piece of work performed by one network involver and is beneficial to another. It also presents the extent to which one network user fulfills the communication requirements of another user. Applications for a service are a group of physical realizations of the service on specific sets of devices and links as shown in Figure 4-2.

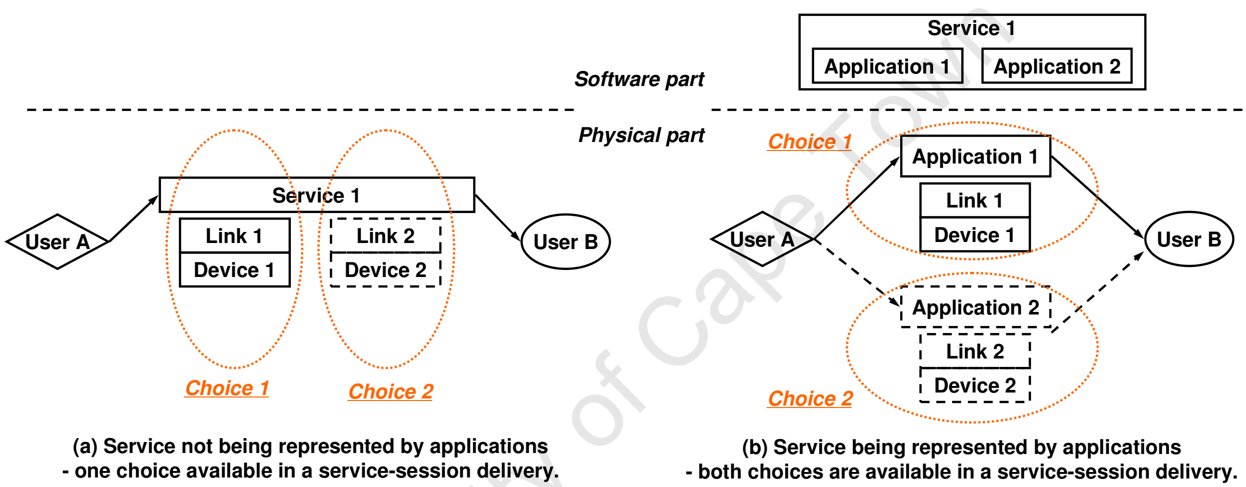


Figure 4-2. Service delivery with or without application implementation.

In Figure 4-2 (a), a service is bonded directly with sets of physical devices and links. When User A starts a service session to User B, he/she can choose a connecting manner either through the pair of Link 1 and Device 1 or through the pair of Link 2 and Device 2. Nevertheless, once the session has started, User A cannot switch to another set of devices and links, whether the original set is available or not. Figure 4-2 (b) illustrates the implementation of a service as two applications in a software environment, with Application 1 physically working on top of the pair of Link 1 and Device 1 and Application 2 on top of the pair of Link 2 and Device 2. In this way, when User A starts the session to User B, the software environment is able to identify the most appropriate application for the session and adjust the choice accordingly. FGN uses the service expression in Figure 4-2 (b) for its flexibility in the determination of optimal connecting manner. Each application in FGN has a unique ID and contains a fixed amount of information.

This ID is temporarily assigned by the network when the application-embedded session starts executing and the network takes back the ID when the session finishes executing.

An application element is a piece of information that FGN uses to represent an application. It generally contains the following parameters and its contents are time insensitive:

- application ID (e.g., “0032”),
- application name (e.g., “VoiceCall_GSM”),
- service type (type of service that the application carries, e.g., “Audio”),
- connecting lifetime (time required to set up a connection for application execution, e.g., “30” seconds),
- application performance (performance requirements on executing the application, concerning bandwidth, packet delay, and bit error rate),
- application availability (availability of the application to a specific session in terms of proper set of device and link, e.g., “Available”, “ToBeAvailable”, or “NotAvailable”), and
- billing manners for the application (e.g., “ChargeToBoth” sender and receiver).

Application elements flow between function modules to assist their individual work. Given the ID of an application, Application Library responds by obtaining all the application-related information from the application database and stores the information in an application element for future use. Session Generator uses the information of an application element, together with that of user elements for the session involvers, to generate a virtual session. To make a proper judgment on the temporarily stored sessions, Session Keeper requires relevant application elements from Application Library.

4.1.3 Session Element

A session element is a piece of information that physically implements a virtual session in FGN. It is composed of a session ID, an application element, several user elements, session connecting lifetime, and the most recent time of refreshing the session information. Session elements stored in Session Keeper have three more compositions to embrace, including session priority, number of changes in user communication status, and session emergency status.

To better understand session element, we need to further explain the concept and generation of virtual sessions. A complete service session in FGN comprises the real-session part in NGN and the virtual-session part in the human-intelligence part, as shown in Figure 4-3:

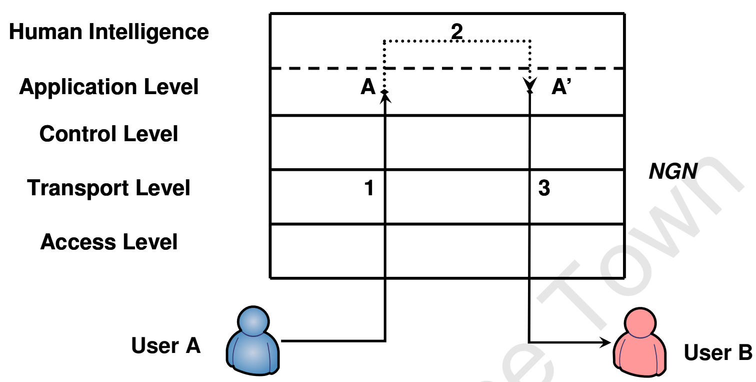


Figure 4-3. Real and virtual properties of a complete session in FGN.

Figure 4-3 illustrates the entire process of carrying out a service session in FGN. A real session starts from User A and continues to point A in the application level of NGN (step 1). The human-intelligence part then creates virtual sessions to simulate the real session in different forms of application requirement and user need. By comparing these virtual sessions, the human-intelligence part proposes several session-delivery solutions and testifies to their success probability (step 2). Finally, the real session will be carried out in the physical-network part (NGN) after receiving the suggested solution from the intelligence part at point A' (step 3). It is feasible to determine an optimal session-delivery solution after having compared several different approaches. It is also easy and flexible to make this determination in the software-based human-intelligence part by using programmable session elements to represent virtual sessions.

Session elements operate over five function modules. Session Registrar first represents a real session as a virtual session in a session element. By referring to the index information carried in this virtual session, Session Generator generates a pair of virtual sessions with complete session information by creating and filling in the content of two related session elements. Session Comparator then compares the two related session elements. If Decision Maker decides to temporarily postpone the session for further examination, the pair of session elements will be stored in Session Keeper.

4.2 System Function Modules

The virtual-user system mainly comprises seven function modules, including Session Registrar, Session Generator, Session Comparator, Decision Maker, Session Keeper, Virtual Personal Profile, and Application Library. Each module independently takes a piece of the job towards imitating human-like intelligence in the network. Figure 4-4 places these modules in three planes and depicts the relations between them.

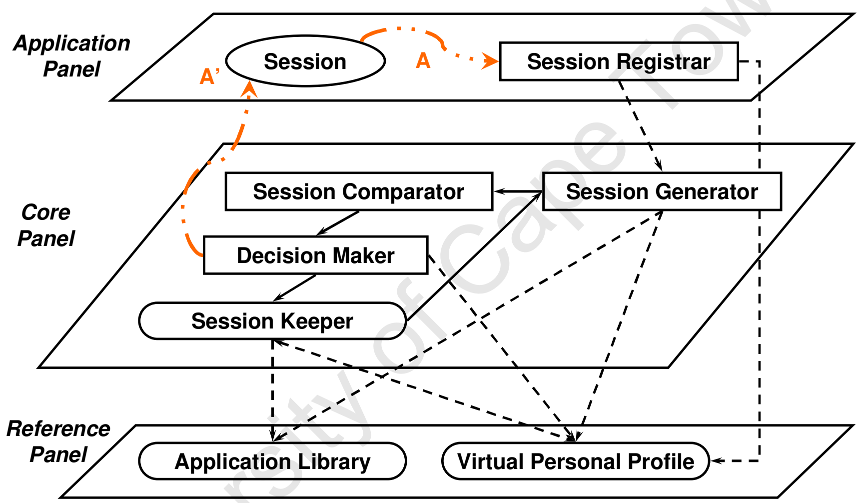


Figure 4-4. Placement and relations of function modules in the human-intelligence part.

In the Application Panel, a network session breaks at point A inquiring an optimal delivery manner and reconnects at point A' after getting feedback from the human-intelligence part. The points A and A' are the same as the points A and A' in Figure 4-3. The module at the end of an arrow requires information from or imposes an action on the module at the beginning of the arrow.

Regarding their collaboration in FGN, the seven modules are placed in three panels with each panel performing a category of similar functions. Among these modules, Session Generator prepares an environment and Session Comparator takes the factual actions for detecting problems and analyzing their causes. Decision Maker follows to make decisions on how to solve the detected problems. Session Keeper provides storage for the temporarily failed sessions to assist with problem solving. These four modules make up the core part of realizing human-like intelligence in the network. Virtual Personal Profile and Application Library work as two reference tables for the four modules in the reference panel. Session Registrar merges into the existing application level of NGN to obtain the information of real sessions.

4.2.1 Session Registrar

Session Registrar (REG) is an operating function module that registers a real communication session in the human-intelligence part. It manipulates mainly user elements.

REG creates an entry for real communication sessions to access the human-intelligence part for intelligence consultation. It conducts the following five actions.

(1) Properly receive a real session from the application level in NGN. To do so, REG needs to have network interfaces compatible with those in NGN and understand the existing policies in NGN. It is preferable to develop REG as a system component in the NGN architecture.

(2) Thoroughly and accurately interpret a real session. On detecting a broken session from the underlying NGN, REG collects from the session the information that counts in applying intelligence and divides the information into three parts: application information, session initiator's communication profile, and session receiver's communication profile. REG expresses the three-part information in a language that all the modules of the human-intelligence part can understand. For example, REG is generating a session element with a fixed format for a real session. If REG finds that the application type of the session is a phone call from a landline, it will fill in the content of "VoiceCallLandline" in the application-type field of the element. It is noted that the session elements generated by SG only contain the index information of a session such as session ID, application ID, and user ID.

(3) Invoke Virtual Personal Profile to fetch user information from the user database. By sending Virtual Personal Profile the IDs of the session initiator and session receiver, REG informs it of the creation of a new session in the human-intelligence part and reminds it to fetch these users' information from the user database if it does not have the information.

(4) Immediately update session involvers' present communication profiles. An incoming session always carries the most recent communication information of the session

initiator. Whenever REG sends the information to Virtual Personal Profile, it resultantly invokes a real-time update of the initiator's information in the human-intelligence part.

(5) Activate Session Generator. REG is mainly responsible for activating Session Generator to generate virtual sessions and sending to it the necessary index information of a session, such as the IDs of the needed application and involved users.

To sum up, REG interacts with both the physical-network part and the human-intelligence part (Virtual Personal Profile and Session Generator). It creates an opportunity for the human-intelligence part to detect communication problems from the physical network.

4.2.2 Session Generator

Session Generator (SG) is an operating function module that simulates real sessions as virtual sessions in the human-intelligence part. It manipulates user- and application-elements.

SG simulates a real service session as a pair of virtual sessions. One virtual session, called the "original session", mimics the originally desired session with the content of all its fields strictly abiding by the original service requirements from the session initiator. However, whether the session is practically deliverable or not is up to the availability of the receiver or the receiving devices. The other virtual session, called the "practical session", simulates the newly created yet practically available session according to FGN's suggestions. The session is certainly deliverable in practice but the delivery manner may or may not be desired by the initiator.

SG uses a pair of session elements to implement such a pair of virtual sessions. (1) When SG generates the "original session", it performs two actions. First, it analyzes what the initiator's original desires indicate. For example, SG has detected that the dialed number is a landline phone number and thus considers the desired application type as "VoiceCallLandline". Second, SG fetches the corresponding information from Application Library and enters it in the corresponding fields of the "original" session element. The information includes the characteristics of the desired application and their corresponding value levels, the desired session receiver's communication details, and the desired system information such as the session-execution time. (2) When SG creates the "practical session", it goes to Application Library to fetch the same application information and goes to Virtual Personal Profile to fetch the related users' latest communication information. Then it fills the information in the "practical" session element. These related users are suggested FGN, instead of being desired by the initiator.

After having generated a pair of virtual sessions, SG invokes Session Comparator to compare them. In addition, SG itself does not store the session pair.

SG may get two types of invoking messages: the one from Session Registrar for generating a pair of virtual sessions for a newly broken communication session, and the other from Session Keeper for resuming an existing session suspended there. There is no difference in content between the two types of messages.

To sum up, SG interacts with Session Registrar, Session Comparator, Session Keeper, Virtual Personal Profile, and Application Library. It prepares an environment for further problem detection in Session Comparator.

4.2.3 Session Comparator

Session Comparator (SC) is an operating function module that compares the performance of two virtual sessions in a pair. SC mainly manipulates session elements.

The principle for session comparison is to check whether a newly created "practical session" is able to meet the session initiator's requirement, which is represented by the "original session". To do so, either of the two sets of leveled requirements in Figure 4-5 needs fulfilling.

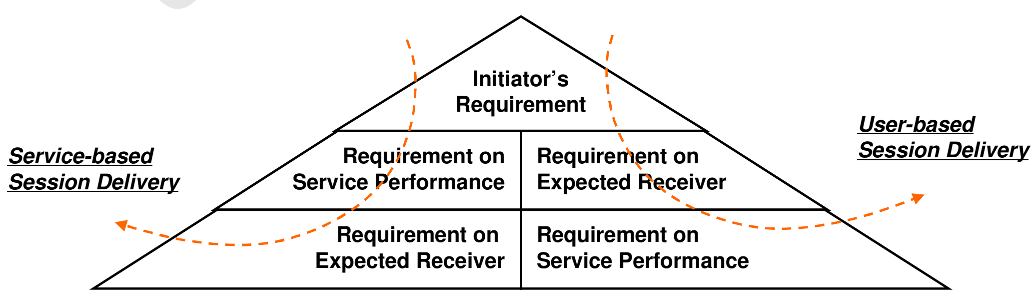


Figure 4-5. Sets of leveled requirements on session execution in a pyramid form.

The higher a requirement sits in the pyramid, the more privilege it has.

The two halves, respectively indicated by the two dashed arrows of the pyramid in Figure 4-5, are two sets of sequential requirements on executing service sessions. A session always meets the requirements in a top-down sequence using a manner from either half. The two resulting session categories must at first and at least fulfill the service requirement by the session initiator at the top of the pyramid. The initiator's requirement then evolves and diverges into two subsequent sets of requirements.

(1) A session meeting the set of sequential requirements on the right half of Figure 4-5 means that the session tries its best to deliver the service to the originally desired service receiver by sacrificing the service performance. We call such a session delivery manner a "user-based session delivery". In this manner, whatever improvement applied to session execution are user-centric. For example, the following mobility mechanisms are user-centric because they aim to improve the expected receiver's accessibility by enhancing his/her mobility:

- device mobility that allows the receiver's device to access different networks,
- session mobility that applies a session to the receiver's different devices,
- service mobility that realizes a service to the receiver by joining several independent sessions, and
- personal mobility that makes the service accessible to the receiver wherever he/she goes.

(2) A session meeting the requirements on the left of Figure 4-5 tries best to maintain the required service performance by sacrificing the originally desired receiver. We call such a delivery manner a "service-based session delivery". Although this delivery manner looks absurd in most cases, it is able to meet the service initiator's requirements better than the other in specific circumstances. For example, a customer initiates a call to the office number of the salesperson Lisa to request general product information. Lisa's officemate Bill is also able to answer such a call. Although the customer dials Lisa's office number, what he/she really cares about is not the actual call receiver but the call service. It is obviously preferable to apply the service-based delivery manner to the session (e.g., forward the call to Bill if Lisa is busy). This manner enables a session to move from the originally desired receiver to another receiver (absolute mobility).

We have also addressed the intrinsic privilege of the session initiator over the session receiver in Figure 4-5. The privilege originates when the initiator starts a session and is therefore ahead of the receiver in being aware of the ongoing session. If the session fails in the network, the initiator will know it and feel disappointed. However, the receiver has no idea of the session at all before the session physically reaches him/her.

FGN adopts a combined approach of the above two types of session delivery. When executing a session, FGN first identifies the most qualified potential receiver using the service-based session-delivery approach and then follows the user-based approach to complete the session. To carry out the user-based approach, SC first compares and establishes all currently available connecting manners for the identified receiver, which may provide the same, better, or worse service performance than desired. It then identifies all potential connecting manners that will become available before the session expires in the human-intelligence part. To do so, the session needs to be suspended somewhere in FGN and be invoked at the correct future time. Service performance in this scenario can also be the same, better, or worse than desired.

SC compares the peered virtual sessions in a pair by comparing their representing session elements. The two session elements have the same format but different values in several specific fields such as the desired receiver and available devices of the receiver. SC compares whether the values of the practical session fully or partially match those of the original session from the same field. It then records the matching results such as whether the two sessions match and, if yes, how much and at what time. SC finally sends all the comparison results to Decision Maker for further decision-making.

To sum up, SC interacts with Session Generator and Decision Maker. It analyzes the causes for the detected communication problems.

4.2.4 Decision Maker

Decision Maker (DM) is an operating function module that determines the final delivery manner for a session. DM is the pivotal module in the virtual-user system. It analyzes the communication status of both the network and users and, based on the analysis of the results, proposes optimal solutions for the identified communication problems. DM mainly manipulates session elements.

DM has two major tasks to perform. One is to choose a session-delivery approach from those shown in Figure 4-5 according to the session initiator's initial desires (section 4.2.4.1). The other is to make apt decisions for the final session-delivery manner (section 4.2.4.2).

4.2.4.1 Delivering Sessions according to User Preference

In theory, DM needs to refer to the initiator's communication profile stored in Virtual Personal Profile for his/her preferred session-delivery approach. Yet we use the relationship type of two session involvers to assist in determining the type of desired session delivery in the virtual-user system for simplicity. Once the preferred delivery manner has been assured, DM will carry out the procedure of session delivery differently according to the determined manner.

(1) If the initiator prefers the user-based session-delivery approach, DM will enumerate all possible connecting manners of the receiver for the session.

(2) If the initiator prefers the service-based approach, DM will enumerate the possible receivers to find the one most capable of maintaining the desired service performance.

(3) If the initiator has no explicit preference for a session-delivery approach, DM will go through all possible connecting manners of all potential receivers to determine the most effective solution for service delivery. To make a proper choice, DM first finds all potential receivers and their individual connecting manners, all of which are theoretically available for the session. DM then evaluates each manner on how it is able to satisfactorily meet the session initiator's requirements as well as take care of the related receiver. DM will finally choose the connecting manner that is best able to fulfill both criteria. The example in Figure 4-6 further explains this theory.

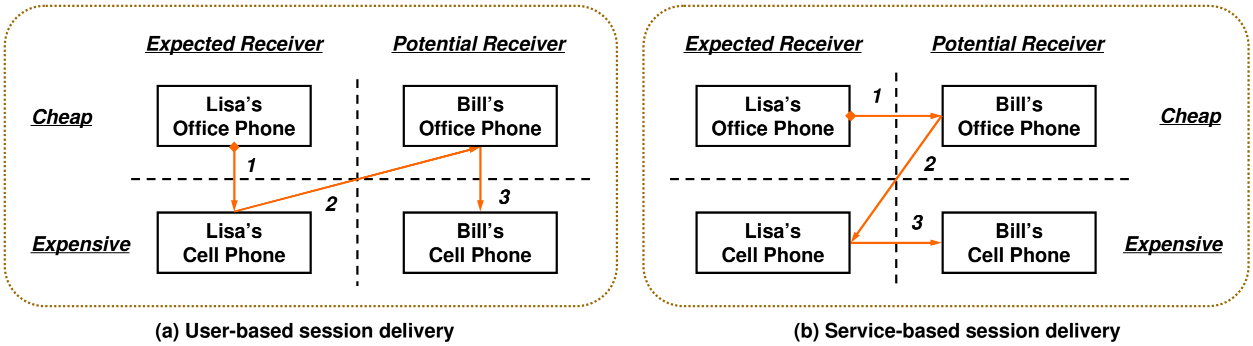


Figure 4-6. Example of identifying a suitable session-delivery manner.

Numbered arrows represent the service-delivery manner in that scenario. The increasing numbers indicate the executing sequence of the manner.

Figure 4-6 exemplifies the user-based and service-based session delivery manners using live communication cases. Both Lisa and Bill are in the office and their individual office and cell phones are available. If their boss calls Lisa about the details of a product, the boss most probably prefers the user-based session delivery because he/she knows that only Lisa is able to answer the questions correctly and thus wants to speak to Lisa directly. If Lisa is personally not available, he/she then has to ask Bill. Whereas, if a customer calls to enquire on general product information, he/she does not mind who picks up the call. It is therefore preferable for the call to first go to low-charge-on-user devices such as the fixed phone, then to the high-charge-on-user devices such as the cell phone, regardless of the owner of these devices. The manner illustrated in Figure 4-6 (a) better applies to the boss's call and that in Figure 4-6 (b) to the customer's call.

4.2.4.2 Carrying out Prioritized Policies for Session Delivery

DM maintains a list of six policies on how to identify successful session delivery. These policies ensure the success of a session and meet the session initiator's service requirements by trying their best to maintain the original service performance. DM organizes the six policies in a descending sequence of most likely satisfying the initiator and, by trying them one by one, DM is able to determine the most satisfactory session-delivery manner as early as possible. These policies sequentially include (1) successfully delivering a session, (2) delivering the session with less satisfactory performance, (3) postponing the session until the expected session receiver becomes available, (4) looking for potential receivers to assist the expected receiver with the session, (5) requesting the expected receiver to learn from potential assistants on session delivery, and (6) failing the session. The following explains these policies.

(1) With reference to a pair of virtual sessions, if DM finds that the practical session matches the original session and is immediately available, it will instruct the physical network to execute the real session according to the originally desired delivery manner.

(2) If DM can find immediately available but less qualified practical sessions, it will refer to user relationship for a suggested session-delivery manner. To do so, DM inquires of Virtual Personal Profile about the relationship between the initiator and the receiver. If the relation demands a user-based service delivery, which means the initiator permits a certain sacrifice of service performance, DM then delivers the session to the expected receiver with modified service performance. Hereof, a less qualified session means that the session cannot completely fulfill the initiator's requirement on service performance. The session performance is either better or worse than the desired. Whether the initiator is willing to accept a better-performance practical session, a worse-performance session, or both depends on the type of relationship.

(3) If DM can only find sessions practically available in the future but still within session connecting lifetime, it will also inquire of Virtual Personal Profile about the relationship between the initiator and the receiver for session delivery. If the relation type requires a user-based service delivery and allows a delay during session set-up, DM will request the human-intelligence part to postpone the session until the expected receiver becomes available. For example, a call receiver is able to receive a phone call that allows a 10-second period for call set-up when he/she moves from a subway without signals to a road with signals within 10 seconds. If DM has decided to postpone the service, it will instruct to temporarily store the session in Session Keeper.

(4) If DM cannot find practical sessions now and in the near future, it will check with Virtual Personal Profile on the initiator's preferred session-delivery approach by user relationship. If the initiator only requires the expected receiver to receive the service (merely user-based session delivery), DM has to fail the session.

However, if the initiator agrees to use service-based session delivery, DM then works out a remedy for the session. DM requests the virtual expected receiver to recommend an assistant receiver who is technically available and personally free for the session. To do so, DM searches the expected receiver's communication profile in Virtual Personal Profile and compiles a list of potential receivers who play an equal role as the expected receiver in the same virtual community. If some potential receiver has the experience of dealing with the same service or dealing with some service in the same service category and the potential receiver is personally free, DM will then recommend the potential receiver as an assistant receiver for the session.

To maintain service security and privacy, DM has to identify a responsible assistant receiver by referring to two social parameters. One parameter is the assistant receiver's absolute trustworthiness on delivering communication sessions and the other is the degree to which the original initiator/receiver trusts the assistant receiver in receiving his/her sessions (section 4.5.1).

Once having approved that some potential assistant receiver is competent for the final receiver, DM will use steps (1), (2), and (3) or a combination of them to continue the session to the finally identified receiver.

(5) If DM finds that the expected receiver has no rules entered in Virtual Personal Profile on how to deal with the session, it will search in the expected receiver's social relations for a trustable assistant who has the rules for processing the session. After having found such an assistant, DM copies the assistant's rules to the expected receiver and requests the expected receiver to carry on the session. In this case, the type and trust degree of the relationship between the expected receiver and the assistant are extremely important.

Once the expected receiver has learnt the method of processing the session, DM will continue the session using the methods in (1), (2), (3), and (4) or a combination thereof.

(6) If DM cannot find any solution to successfully deliver the session even after having tried all the above manners, it instructs the physical network to fail the session.

The human-intelligence part takes six respective actions to carry out the decisions from (1) to (6), including *deliver*, *force*, *postpone*, *help*, *learn*, and *fail*.

To sum up, DM interacts with Session Comparator, Session Keeper, Virtual Personal Profile, and the physical network. It maintains the policies of proposing optimal solutions for session delivery in a thorough way and uses them to determine the most reasonable session-delivery manner.

4.2.5 Session Keeper

Session Keeper (SK) is an operating function module as well as a storage function module. It mainly manipulates session elements.

SK stores temporarily postponed sessions and assists in delivering these sessions at an appropriate time. It physically contains a relational-model database to keep static records of sessions and logically maintains the session-processing mechanisms that deliver the sessions according to their priorities. SK carries out many actions to realize those functions.

Firstly, SK stores suspended sessions as session elements. Each session element comprises the original session ID for consistency and identification purposes, desired application information, session initiator's information, and designated session receiver's information.

Secondly, SK chooses the best time to invoke the regeneration of those ready-for-processing sessions. The best time is as soon as the suspended session is physically ready for regeneration. SK calls on three parallel threads to ensure the realization of such an effective choice. (1) SK frequently calculates the priority of each stored session and sends the session with highest priority to Session Generator after each calculation. Session priority determines how fast a service session can reach the point of being processed. This priority is able to reflect the real-time deliverability of the session by considering the session's set-up time, its storing time in SK, application type, the receiver's current communication status, relation of the initiator and the receiver, and session emergency status. The calculation of session-process priority needs a great amount of work (section 4.5.2). (2) To correctly calculate the process priority of a session, SK regularly contacts Virtual Personal Profile on the status of the receiver. Any change in the receiver's most current communication status has a significant effect on the calculation of session priority. (3) Meanwhile, SK is ready anytime to receive messages from Virtual Personal Profile about the change it has detected in the receiver's current communication status.

SK is very capable of dealing with high-volume real-time communication information. (1) SK is able to store a large amount of dynamic session information by maintaining an independent database. (2) SK is adaptive to the change of time because the session information includes time-sensitive user information such as the user schedule. (3) SK is able to immediately produce useful session information whenever appropriate. (4) SK takes the initiative to exchange information with another module such as Virtual Personal Profile. (5) SK is able to trigger other modules such as Session Generator and can itself be triggered by another module such as Virtual Personal Profile.

We bring Dynamic Service Session Database (DSS-DB) into FGN to work as SK. DSS-DB is a network database designed to store the real-time changing communication information [82]. It therefore has several outstanding advantages to operate as SK. (1) DSS-DB is born to store dynamic information and thus has the ability to store real-time service sessions. (2) DSS-DB is capable of managing a wide range of communication services including data, audio, and video. (3) DSS-DB is able to calculate session-processing priority. (4) DSS-DB communicates real-time with the external environment and the communication results affect the database output at any moment. Nevertheless, unlike DSS-DB listening to the communication status of both the network and users, SK listens to user communication status in the exterior environment with the assumption that the physical network is always ready for communication.

To sum up, SK interacts with Decision Maker, Session Generator, Virtual Personal Profile, and Application Library. By keeping the on-hold sessions in SK, FGN is able to fulfill one of its intelligence policies – postponing temporarily postponed sessions for a reasonable period. In this way, FGN has better control of the entire network's traffic.

4.2.6 Virtual Personal Profile

Virtual Personal Profile (VPP) is a storage function module that keeps both fixed and dynamic communication information of network users. It mainly manipulates user elements.

VPP uses a user-element list to keep the communication profiles for the network users who are currently involved in the ongoing sessions of the human-intelligence part. Because each user element in VPP acts as the virtual part of a network user, the entire VPP works as a virtual-user community where the virtual users exist and present themselves to each other. VPP is adaptive due to the time-sensitive feature of its user elements.

VPP interacts with many function modules to exchange user information. (1) On being notified by Session Registrar of a new session, VPP first checks whether it has the information of all session involvers on its list. If there is, VPP simply refreshes the involvers' current communication profiles using the information from Session Registrar. If there is not, VPP fetches the involvers' communication information from the user database, generates a few pieces of corresponding user elements, and stores the information for further use. (2) Soon after, when Service Generator needs to generate virtual sessions, it will fetch the session involvers' information from VPP by copying the corresponding user elements. (3) Whenever VPP detects any abrupt change in the involvers' current communication status, it takes proactive action to inform Session Keeper of the changed user status by sending the newly updated user elements. Such action applies to a scenario where a network user is involved in more than one session at a time. The user's current communication status in an ongoing session can be affected by its participation in another session. For example, Lisa's schedule shows that she is in a weekly meeting from 2:00pm to 3:00pm every Wednesday, whereas the meeting is temporarily cancelled this week. When a call goes to Lisa's office phone at 2:40pm, FGN generally fails the call according to the schedule. If Lisa made a phone call at 2:30pm using her office phone and this action notified FGN of Lisa's availability on her office phone at that time, FGN can then properly direct the incoming call to Lisa's office phone. (4) VPP also provides Decision Maker with the information of the session involvers' relationships when Decision Maker needs it to help with decision-making.

To sum up, VPP interacts with Session Registrar, Session Generator, Decision Maker, and Session Keeper and works as a reference table for these modules. These modules turn to VPP whenever they need user information.

4.2.7 Application Library

Application Library (ALIB) is a storage function module that keeps static application information. It mainly manipulates application elements.

ALIB maintains a reference table of all types of applications that are described by a fixed number of predetermined characteristics. It distinguishes between these applications by assigning an exclusive combination of values for all characteristics to each application.

ALIB interacts with many function modules to exchange application information. (1) At the set-up stage of a session in the human-intelligence part, ALIB fetches all application information from the application database and simply keeps them there for future use. ALIB regularly updates the information to ensure it has the most recent. The updating period is comparably long, for example, one week. (2) ALIB provides Session Generator with application information for generating virtual sessions. (3) It also provides Session Keeper with application information for calculating session priority.

To sum up, ALIB interacts with Session Generator and Session Keeper. These operating modules refer to ALIB for needed application information.

4.3 Major Interworking Mechanisms

Now that we are clear on the individual tasks and mechanisms of each function module, we address the major interworking mechanisms between these modules to prepare for describing data flows of the virtual-user system. Figure 4-7 illustrates an overall view of the interworking mechanisms:

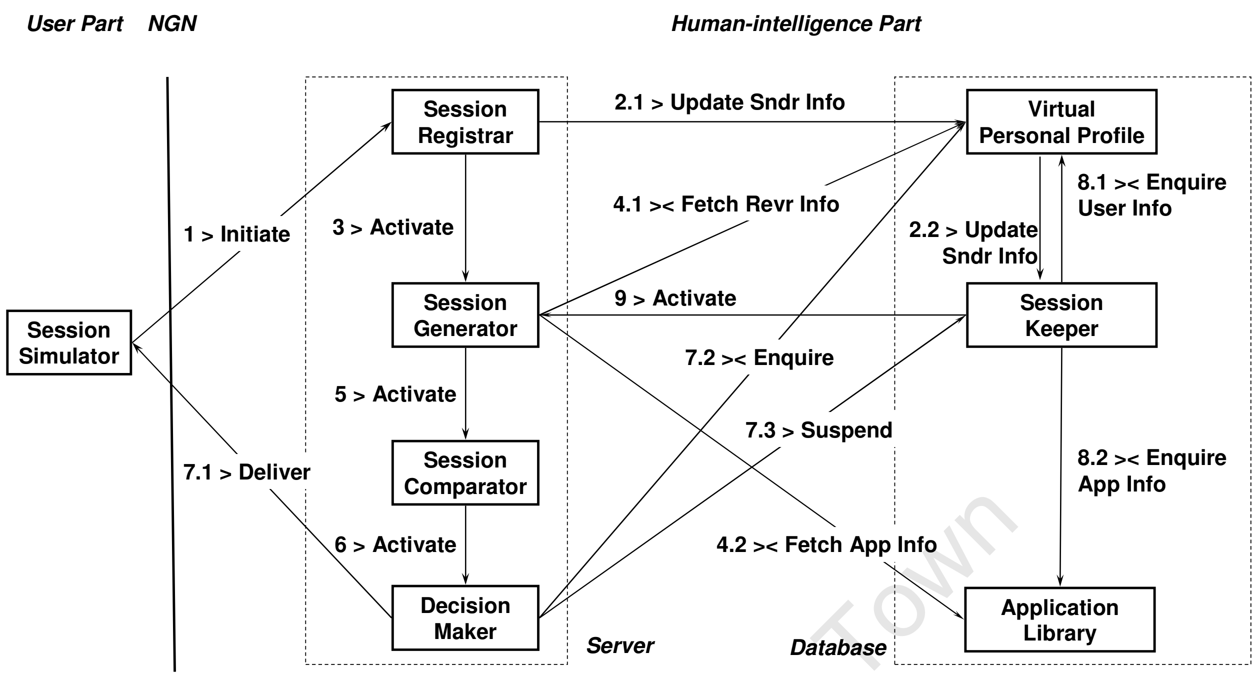


Figure 4-7. Interworking mechanisms in the functional design of the virtual-user system.

Mark ">" indicates that the module where an arrow begins sends a requirement or a demand to the module that the arrow points to. Mark "><" indicates that the module where an arrow begins obtains information from the module that the arrow points to. "Sndr", "Rcvr", "App", and "Info" respectively stand for the items Sender, Receiver, Application, and Information.

Figure 4-7 exhibits how the modules of the virtual-user system cooperate to complete the task of implementing human-like intelligence on a session in the network. To generate the input and deliver the output of the system, Figure 4-7 introduces a new function module called Session Simulator that simulates a service session with needed application- and user-information.

Session Simulator initiates a communication session to Session Registrar (step 1). The Registrar first updates the session initiator's information in Virtual Personal Profile (step 2.1) and, when necessary, further updates the user information in Session Keeper (step 2.2). Meanwhile, it activates Session Generator (step 3). The Generator fetches user communication information from Virtual Personal Profile (step 4.1) and application information from Application Library (step 4.2) to generate a pair of virtual sessions and then activates Session Comparator to compare these sessions (step 5). Thereafter, the Comparator sends comparison results to Decision Maker for decision-making on session delivery (step 6). Decision Maker delivers the session to Session Simulator (step 7.1), enquires of Virtual Personal Profile about its suggestions (step 7.2), or suspends the session in Session Keeper (step 7.3). Session Keeper obtains user- and application-information respectively from Virtual Personal Profile and Application Library to calculate the priorities for the suspended sessions (steps 8.1 and 8.2), thus activating Session Generator to reconstruct the sessions when necessary (step 9).

To be able to effectively communicate with each other as described above, these modules need to perform two types of action protocol: reactive- and proactive-protocols. The two protocols have regulated the in-all interacting manners between the modules in the virtual-user system. (1) In the reactive protocol, a module acts on getting an incoming request from another module. For example, only on receiving an “activate” instruction from Session Registrar, Session Generator starts composing a service session (step 3). The Generator will not take the initiative to regularly inquire of the Registrar about when it is supposed to generate sessions and what the prerequisite information is. (2) In the proactive protocol, a module initiatively enquires of another module for relevant information to determine whether it needs to act. To real-time update the priorities for the suspended sessions, Session Keeper uses the proactive protocol to initiatively fetch information from Virtual Personal Profile and Application Library at regular intervals (step 8.1). The latter two storage modules have no way to actively detect the temporarily stored sessions in the Keeper and thus cannot determine useful information for them. However, in a specific case where Virtual Personal Profile calls on Session Keeper to recalculate the session priority after receiving an update of user information from another session, the Keeper does react to the request message using reactive protocol (step 2.2).

Another point is that two modules must agree on the same rules or speak the same language in order to interact. For example, Session Comparator compares a pair of virtual sessions and sends the comparison results to Decision Maker. To correctly interpret the comparison results, the Maker has to agree with the ways that the Comparator expresses them.

4.3.1 Illustrating the Finite State Machine of a Session

Finite State Machine (FSM) illustrates system behaviour by modelling the transactions between a finite number of states ([83]). Hereof we adopt the FSM to exhibit the state change of a session in the virtual-user system (Figure 4-8). The change of session states illustrates the interactions between the function modules in Figure 4-7 in a different manner. In its life cycle in the virtual-user system, each session goes through most function modules in a specific sequence (as a pipelining). If one module has completed its part of the job on the session and successfully passed the session to the next module, the session finite state will be updated at the same time. Namely, the change of session finite states exhibits the functioning of individual modules as well as the successful interworking between them.

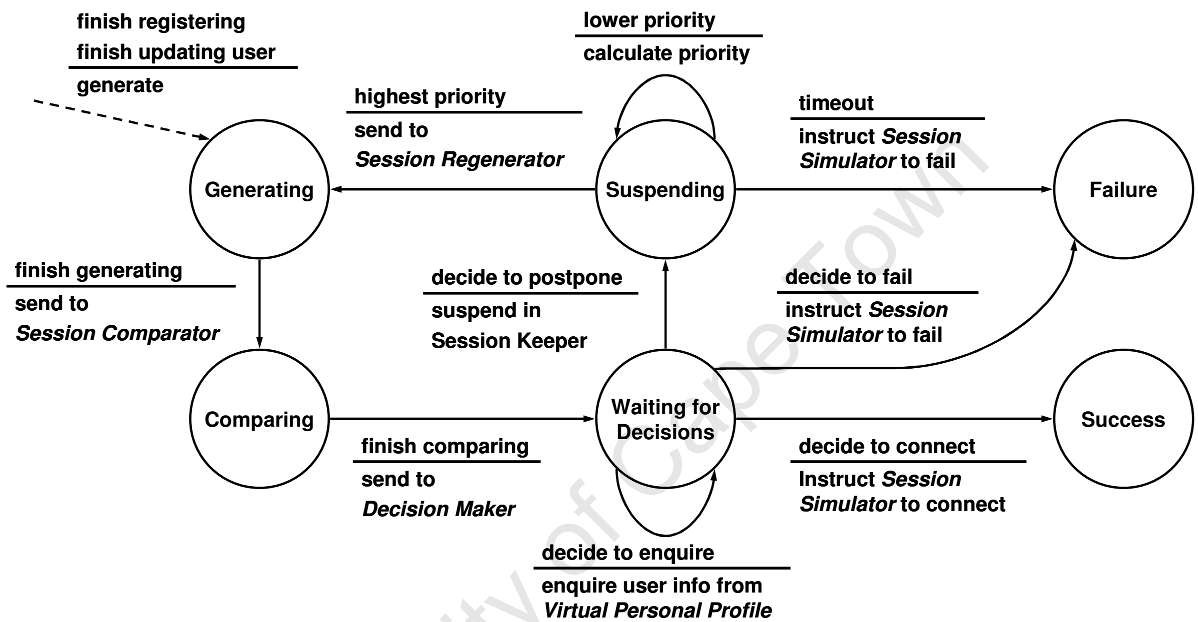


Figure 4-8. Finite states of a session in the virtual-user system.

Circles indicate the states of a session in the human-intelligence part after it has registered. Arrows indicate the change of the session from one state to another. The actions causing the change are shown above the horizontal line and the actions taken when the change occurs are below the horizontal line.

Figure 4-8 shows how the states of a service session change in the virtual-user system. The executing states “Generating”, “Comparing”, “Waiting for Decisions”, and “Suspending” respectively take place in Session Generator, Session Comparator, Decision Maker, and Session Keeper. The resulting states “Failure” and “Success” occur in Session Simulator.

The change of a session from one state to another implies a transaction between two connected function modules. The initial session-state “Generating” means that Session Registrar has instructed Session Generator to construct virtual-session pairs for the principal session. The state “Comparing” indicates that Session Generator has passed the generated virtual-session pairs to Session Comparator. The state “Waiting for Decisions” implies that Session Comparator has sent the session-comparison results to Decision Maker where the principal session awaits a decision on delivery manner. From then on, the next session state can be any of the three possible states - “Success”, “Failure”, or “Suspending”, which respectively indicate that the Maker will instruct to deliver, fail, or postpone the session. The first two states indicates that Decision Maker has sent the session with decision-making results to Session Simulator for further execution, and the last state implies that the Maker has passed the session to Session Keeper for intelligence rescue. In the Keeper, the session faces three possible next states as well. If session state changes to be “Generating”, Session Keeper has bounced the session back to Session Generator for session reconstruction. If the session state remains as “Suspending”, the Keeper has no interaction with other modules. If the state turns into “Failure”, the session has reached its lifetime limitation and the Keeper sends the session to Session Simulator for execution.

4.4 Flow Charts of Different Communication Scenarios

The following set of flow charts will illustrate in detail how the function modules of the virtual-user system interact with each other to recognize occurring session problems and, under different circumstances, choose the most appropriate manner to deliver the problematic session. Figure 4-9 and Figure 4-10 first depict the flows that the virtual-user system detects and analyzes communication problems.

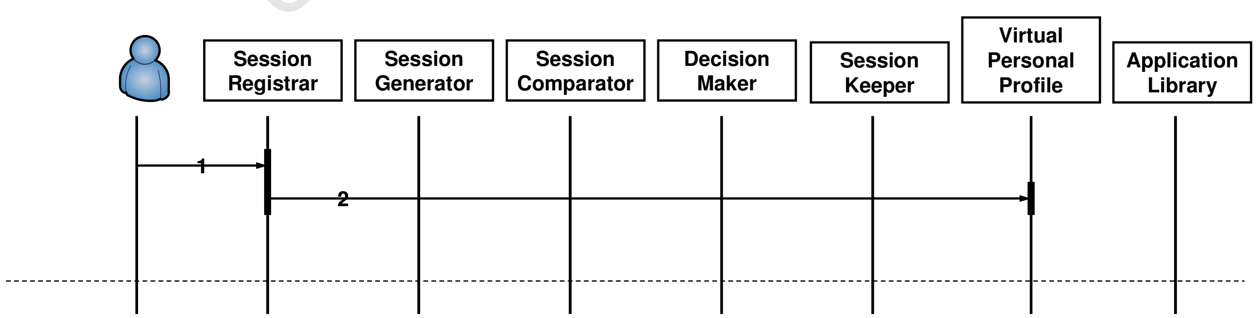


Figure 4-9. Problem detection in intelligence-implementing procedure.

Session initiator sends a service session to FGN and the session breaks at Session Registrar to activate an advanced session process in the human-intelligence part (step 1). Session Registrar then updates the service initiator's most recent communication status using the information in the broken session (step 2). In fact, the human-intelligence part does not actively detect network problems, but the actions of breaking a session and sending the session to Session Registrar for investigation have created a chance for FGN to discover the potential problems.

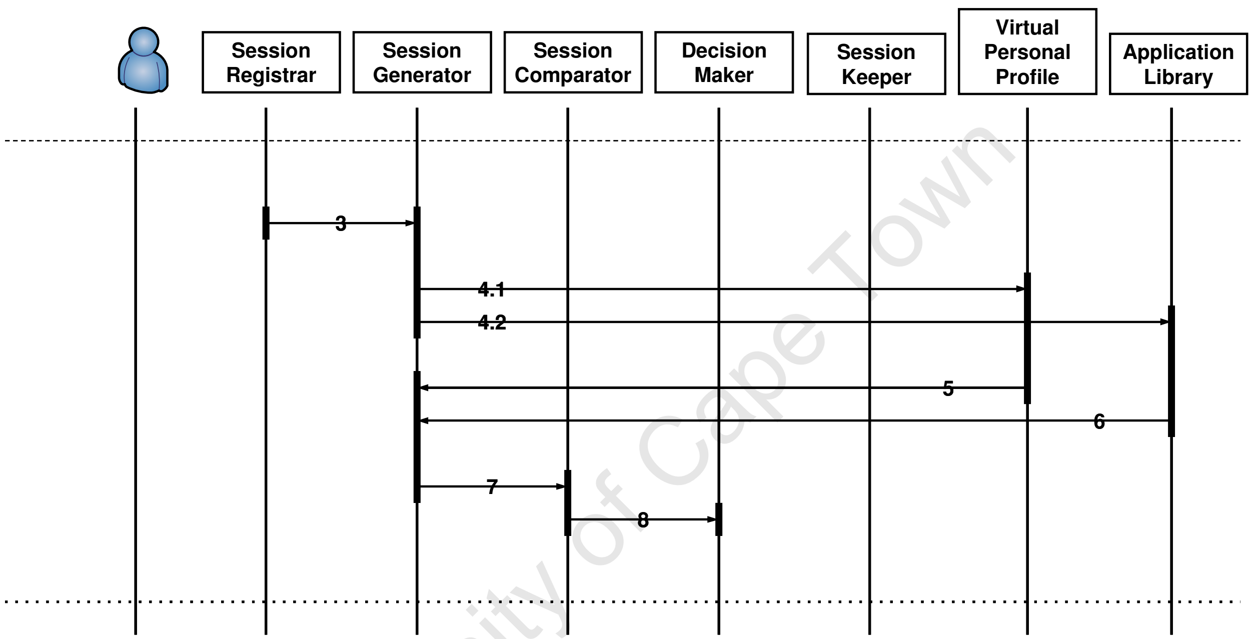


Figure 4-10. Problem analysis in intelligence-implementing procedure.

The virtual-user system then performs a series action to facilitate problem analysis. Session Registrar first activates Session Generator (step 3) to diagnose the problems. Upon the activation request, Session Generator obtains the information of the session involvers' communication information (steps 4.1 and 5) and application information (steps 4.2 and 6) respectively from Virtual Personal Profile and Application Library. It then generates a pair of virtual sessions, with the original one describing the genuine requirements from the session initiator and the practical one implying the expected receiver's current communication status.

Session Generator then sends this virtual-session pair to Session Comparator for comparison (step 7). The comparison results include (1) whether the practical session is able to meet the original requirements and, (2) if able to, whether the practical session completely or conditionally meets the requirements. Thereafter, Session Generator sends the comparison results to Decision Maker to call on its action and assist it in making a decision on how to continue the session (step 8).

From this step on, Decision Maker has six optional ways to deliver a session according to the six decision-making policies (section 4.2.4.2). The following sections illustrate these ways separately to show how Decision Maker makes reasonable judgments and decisions in each.

4.4.1 Procedure of Session Break and Reconnect

If the comparison results show that the practical session can fully meet the original requirements of the session initiator, Decision Maker instructs NGN to successfully deliver the session to users (step 9) as shown in Figure 4-11.

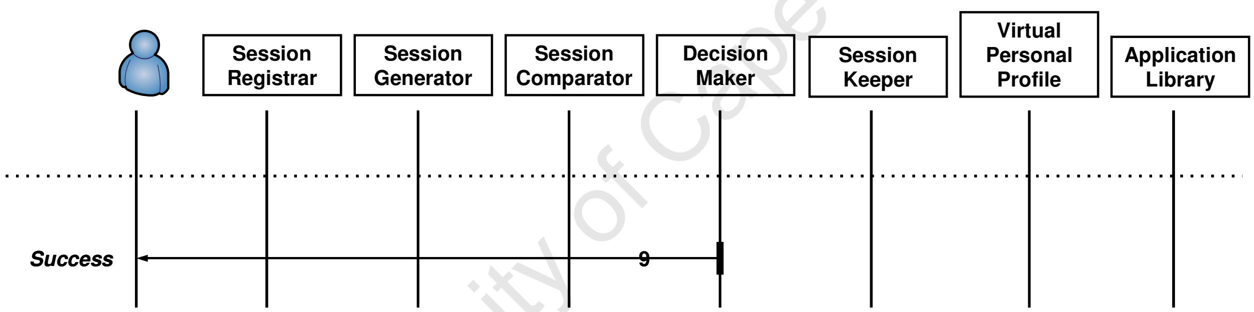


Figure 4-11. Decision of successful delivery in intelligence-implementing procedure.

4.4.2 Procedure of Session Break, Enquire, and Reconnect

As shown in Figure 4-12, if the comparison results show that the session conditionally meets the original requirements, Decision Maker contacts the virtual initiator in Virtual Personal Profile to establish whether the session initiator agrees to deliver the session with conditions (steps 9 and 10). If the initiator chooses the user-based delivery manner and the application requires real-time, Decision Maker instructs NGN to carry on the session with less satisfactory service performance (step 11).

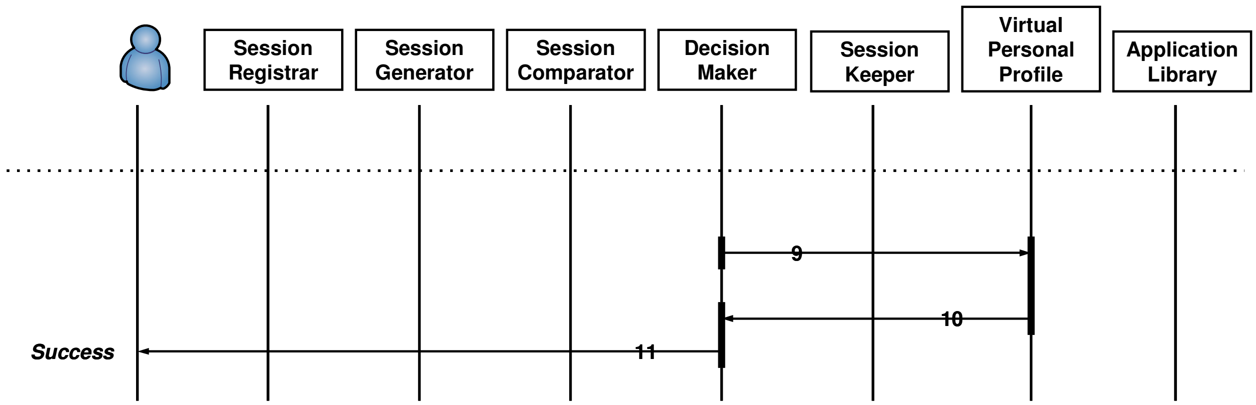


Figure 4-12. Decision of asking for permission in intelligence-implementing procedure.

4.4.3 Procedure of Session Break, Wait, and Reconnect

If the session initiator prefers the user-based session-delivery approach and it would also like to keep the original service performance, Decision Maker will suggest postponing the session until the expected receiver becomes available within a valid period. Flows concerning the postponing manner are shown in Figure 4-13:

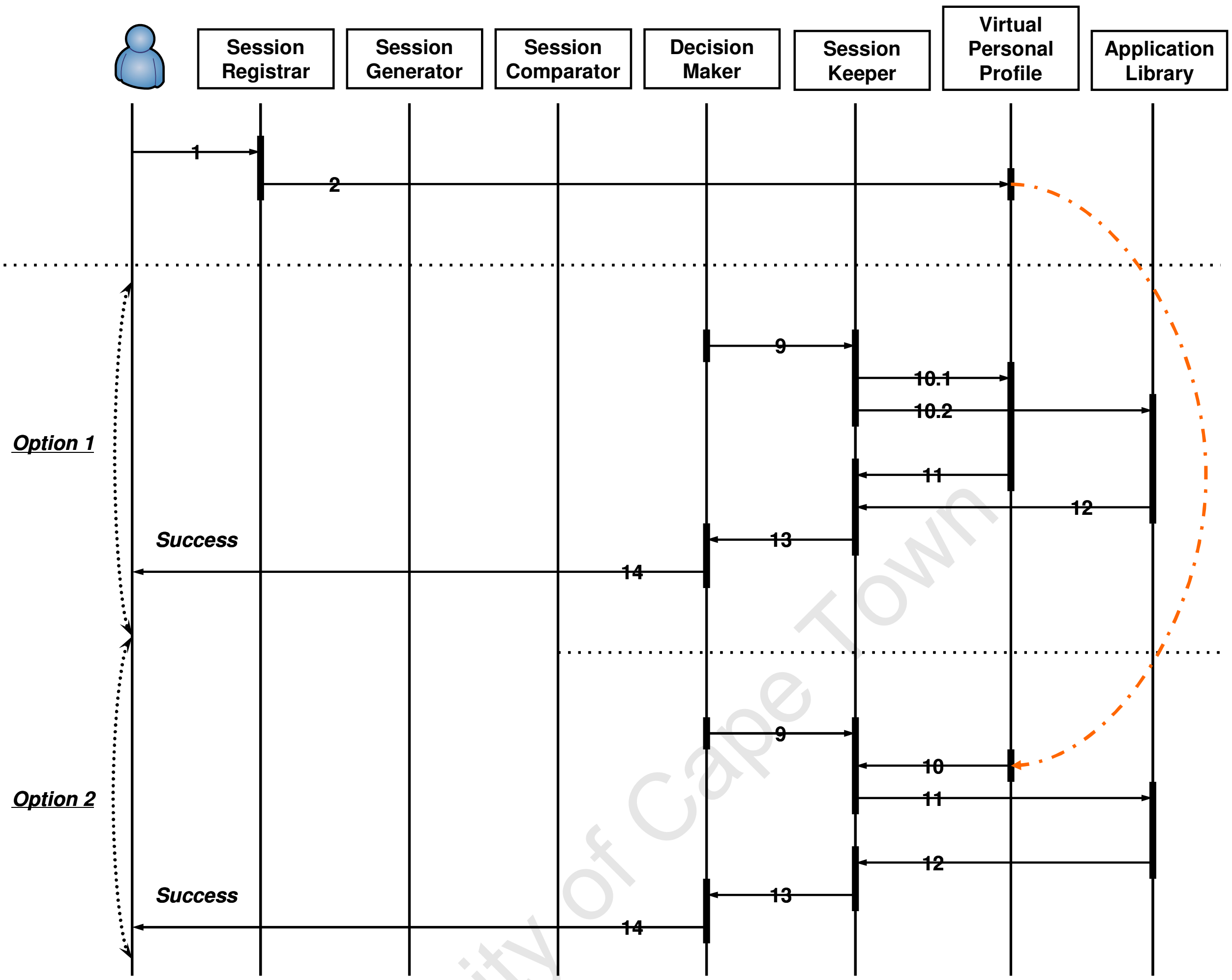


Figure 4-13. Decision of session postponing in intelligence-implementing procedure.

As shown in Figure 4-13, if the virtual initiator agrees to postpone the session for a while, Decision Maker instructs to temporarily store the session pair in Session Keeper (step 9). Thereafter, Service Keeper either frequently asks for the latest information about session involvers and application from Virtual Personal Profile and Application Library individually (option 1, steps 10.1 and 10.2) or waits for an invoking message that conveys the changed information of service involvers from Virtual Personal Profile (option 2, step 10). Eventually, Service Keeper will get a chance to send the session to Decision Maker for decisions (step 13) and the system delivers the session according to the final decision (step 14).

4.4.4 Procedure of Session Break, Reroute, and Forward

If the service initiator chooses to use the service-based session-delivery approach, FGN should be able to identify and suggest a trustable practical receiver based on the recommendation of the expected session receiver, as shown in Figure 4-14:

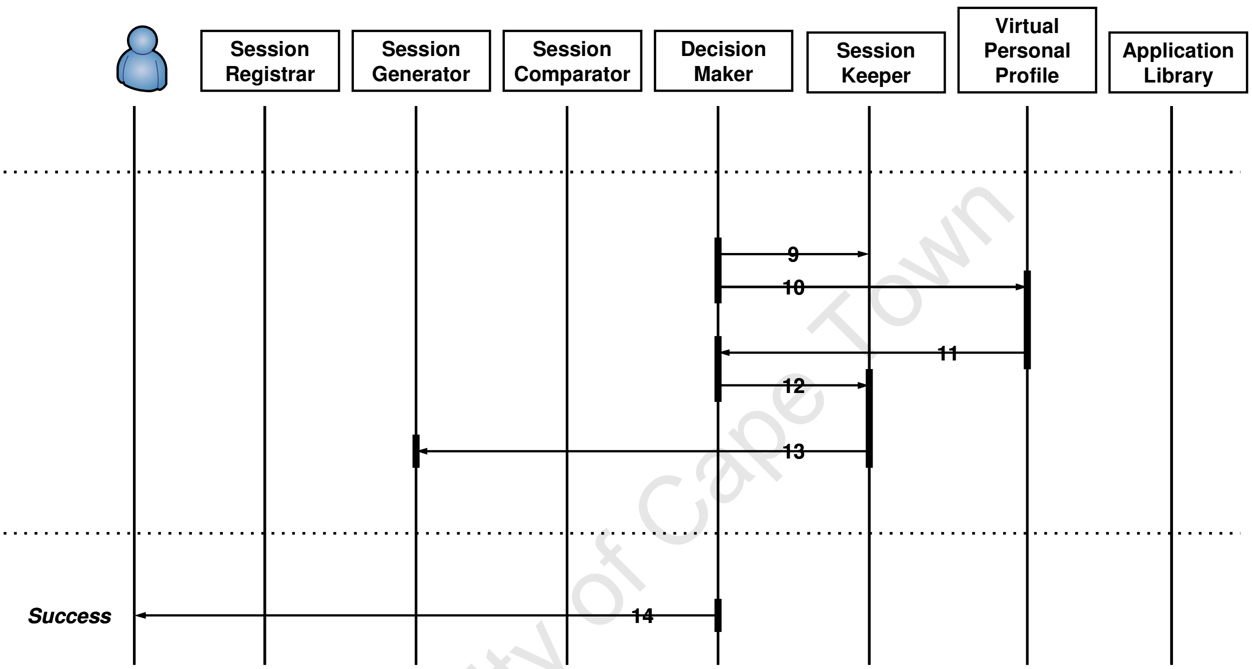


Figure 4-14. Decision of turning to third party in intelligence-implementing procedure.

In Figure 4-14, after Decision Maker has kept the session in Session Keeper (step 9), it asks the virtual expected receiver in Virtual Personal Profile to recommend a practical receiver for the session (step 10). On receiving the recommended receiver's communication information (step 11), Decision Maker will request Session Keeper to send the suspended session to Session Generator (steps 12 & 13) for session regeneration and comparison. The rest of the work will follow the procedure of the steps 4.1 to 8 in Figure 4-10.

4.4.5 Procedure of Session Break, Reroute, and Reconnect

A specific scenario could be that Decision Maker does not know which delivery manner to use because the expected service receiver has no rules to deal with a specific type of session. Decision Maker will suggest that the expected receiver learn from its trusted relations on further session processing. This procedure shares the same flow sequence as that in Figure 4-14 but the content of some flows differs. Decision Maker requests the virtual expected receiver to learn from other virtual users (step 10), expecting to receive the information concerning the expected receiver (step 11).

4.4.6 Procedure of Session Break and Fail

If the comparison results show that the session absolutely cannot meet the original requirements of the session initiator, Decision Maker indicates NGN to fail the session (step 9) as shown in Figure 4-15.

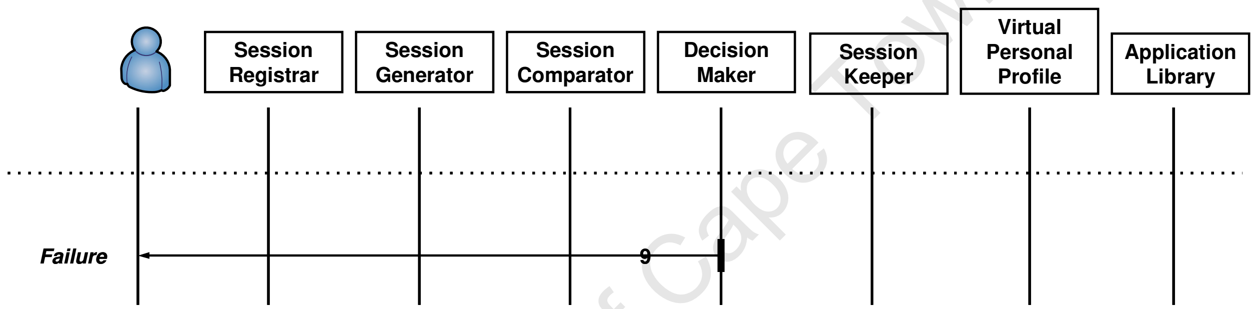


Figure 4-15. Decision of session failure in intelligence-implementing procedure.

4.5 Key Parameters and Their Calculations

Three numeric parameters facilitate manipulating or maintaining the individual and interactive mechanisms (section 4.3). A network user's absolute trustworthiness assists the network in identifying available capable human resources when needed (section 4.5.1). The priority of a suspended session determines the most appropriate time to reconstruct the session (section 4.5.2). The emergency status of a network user represents the status of the sessions with which he/she is involved and thus affects the calculation of the session priority (section 4.5.3).

4.5.1 Quantifying Trustworthiness of Network Users

A communication session generally happens between an initiator and a receiver. To begin a session, the initiator first investigates its trust relationship with the receiver and then uses the relationship to determine whether and, if so, how to communicate with the receiver. The current communications network assumes that, if an initiator starts a session to a receiver, it trusts the receiver in handling the session. However, in the proposed three-user communication scenario (section 3.3.2.2) where the final receiver might be some third-party user suggested by FGN instead of the originally desired receiver, the initiator will have a problem trusting the suggested user either personally or technically in handling the session. An actual and referable value is necessary to help the initiator judge whether to trust the suggested user.

An absolute trustworthiness that describes a social entity's intrinsic trustworthy characteristic ([84]) helps determine how trustable a virtual network user is in the communications network. As absolute trustworthiness does not rely on the user's relationship with any other specific user, its quantified value can be used by various virtual users in different sessions. A virtual user can exhibit trustworthiness in a communication session by showing the ability to process the session, its social and physical availability for the session, and how much it would like to be involved in the session.

To reflect a user's true trustworthiness and consequently provide a valuable trustworthiness value for other users' reference, the network needs to real-time track the user's actual trustworthiness that varies with his/her communication behaviors. This real-time trustworthiness comprises of four dynamic compositions: Trustworthiness on Time (TR@Time), Trustworthiness on Neighbor (TR@Neighbor), Trustworthiness on Domain (TR@Domain), and Trustworthiness on Self-recommendation (TR@Self) [84]. TR@Time shows the history of a user's long-term trustworthiness and the rest cooperatively imitate an external environment that affects the trustworthiness. TR@Time and TR@Neighbor collaborate to determine a user's capability to handle a communication event. TR@Domain determines a user's availability for a specific type of event, while TR@Self gives the user a chance to declare how enthusiastic he/she is about the event.

A user's absolute trustworthiness at time t is expressed as $TR_t()$ and the corresponding compositions of TR@Time as $TR_{t-}()$, TR@Neighbor as $TR_n()$, TR@Domain as $TR_d()$, and TR@Self as $TR_s()$. We confine the value of each $TR()$ in the range[40] for easy quantification.

(1) TR@Time is defined in [84] as a user's weighted trust fidelity in the most recent time periods and is computed by summing up its respective trustworthiness values in a finite number of fixed-length time periods. The closer a time period is to the computing moment, the more it influences the user's trustworthiness. This is because the most recent time periods truly reflect the user's capability of dealing with communication events.

For an easier collection and calculation of data, FGN simply takes the trustworthiness values that a user has for the most recent communication events as a series of TR@Time values. These values reflect the user's experience in handling communication events in the most recent past, although the breaks between any two of them differ.

TR@Time is then expressed as $TR_{t-}(t) = [wt_1 \ldots wt_M] \times \begin{bmatrix} TR_t(t-1) \\ \ldots \\ TR_t(t-M) \end{bmatrix}$, where wt is the weighting attached to each event considered, $TR_t(t)$ is the absolute trustworthiness value obtained from executing the i^{th} communication event ($i \in [1, M]$), and M is the number of most recent events taken into calculation. This expression abides by two principles. The first is that the closer the events are to the computing moment, the more weighting they have. The second is that the sum of the weightings should be a unit to confine the total trustworthiness value within the range. The first principle represents the event weightings as $wt_i = (A/\mu) \cdot e^{-i/\mu}, i \in [1, M]$ and the second principle as $\sum_{i=1}^{M} wt_i = 1$. We can get $A = 1 \Big/ \sum_{i=1}^{M} e^{-i}$ and transform the TR@Time in Eq. (1) by assuming $\mu = 1$:

[40] The range refers to a natural number between zero and a hundred, often expressed as "[0, 100]".

$$TR_{t-1}(t) = \sum_{i=1}^{M} \left\{ \left[e^{-i} \Big/ \sum_{j=1}^{M} e^{-j} \right] \times TR_t(t-i) \right\} \qquad (1)$$

(2) TR@Neighbor is defined in [84] as a user's weighted trust fidelity with its socially related neighbors, that is, how much the user's social neighbors trust him/her to process an event. TR@Neighbor is computed from the evaluation values that the user gets from clusters of his/her neighbors. A cluster assembles the neighbors with a similar frequency of contacting the user. The closer the neighbors of a cluster are to the user, the more the user is worth of their appraisal.

FGN simply sums up the trust degrees that all the user's neighbors give to it. Each trust degree is weighted according to the relationship closeness of the user and the neighbor. Although this definition of TR@Neighbor has simplified the calculation, it has lost the ability of tracking how the change of a relationship affects communication events. For example, two users' relationship-type is "friend". In theory ([84]), the two users' TR@Neighbor value increases when they contact each other more times whereas the value remains the same in FGN because of their fixed relationship type – "friend". It is suggested that TR@Neighbor only be applied to a small neighborhood because it needs to determine the trust degree that a user obtains from each of its neighbors. The more neighbors the user has, the more complex the determination is.

TR@Neighbor is expressed as $TR_n = [wn_1 \ldots wn_N] \times \begin{bmatrix} TR_n(n_1) \\ \ldots \\ TR_n(n_N) \end{bmatrix}$, where wn is the weighting to each neighbour according to their relationship, $TR_n(n)$ is the trust degree to which the user's neighbour trusts the user, and N is the number of neighbours. The sum of these weightings should be a unit, i.e., $\sum_{i=1}^{N} wn_i = 1$, to confine the TR@Neighbor value in the range. Then we have

$$TR_n(n) = \sum_{i=1}^{N} \left[wn_i \times TR_n(n_i) \right] \qquad (2)$$

(3) TR@Domain is defined as a user's availability for communication events in a specific domain. A user's availability means physical availability as well as reputation in a domain. The domain reputation depends on the user's social position and social relations with others in the domain and it affects TR@Self. Because the service that an event conveys has different trust fidelities in different domains, these domains have a different effect on the event with regard to reputation at a certain moment. For example, a hard worker has a very high TR@Domain value in the office but a low value at home, because he/she spends more time in the office than at home.

TR@Domain is expressed as $TR_d = [wd_1 \ldots wd_P] \times \begin{bmatrix} TR_d(r_1) \\ \ldots \\ TR_d(r_P) \end{bmatrix}$, where wd is the weighting of the user's trustworthiness in each domain, $TR_d(r)$ is the trusting value given by that domain, r is the user's domain reputation, and P is the number of domains in which the user is involved. The domain weightings determine how relevant the domain is to a session. Assume each relevant domain has an equal effect on all sessions and $\sum_{i=1}^{P} wd_i = 1$, then $wd_i = 1/P, i \in [1, P]$. Assume $r \in [0.0, ..., 1.0]$ and then TR@Domain can be expressed as in Eq.(3) with θ as a coordinating factor for TR@Time and TR@Neighbor:

$$TR_d = 1/P \cdot \sum_{i=1}^{P} r_i^{[\theta \cdot TR_{t-} + (1-\theta) \cdot TR_n]} \tag{3}$$

(4) TR@Self reflects whether a user recommends him/herself to conditionally accept a communication event or not. The parameter is determined by the willingness to receive the event and the domain reputation of the user. TR@Self concerns whether the user does or does not recommend him/herself for the event. Only when the user's social reputation is greater than a threshold value, is he/she allowed to recommend him/herself for the event, so TR@Self can be expressed as in Eq. (4) with D as the reputation threshold and S as the maximum recommendation value:

$$TR_s = \begin{cases} 0, & r \in [0, D) \\ [(1-S)/(1-D)] \cdot r, & r \in [D, 1] \end{cases} \quad where \begin{array}{l} D \in (0.0, 1.0] \\ S \in [0.0, 1.0] \end{array} \tag{4}$$

Because a real user cannot manually input a TR@Self value for each session, FGN uses the session's physical availability to work on behalf of TR@Self. If the physical network is available, TR@Self is valued unity; otherwise, TR@Self is zero.

(5) A user's total trustworthiness at moment t can be written as

$$TR(t) = \alpha \cdot TR_{t-}(t) + (1-\alpha) \cdot \{[\beta \cdot TR_n(t) + (1-\beta) \cdot TR_d(t)] \cdot TR_s(t)\} \tag{5}$$

Parameters α and β are the affecting factors that confine the reliance of trustworthiness to the user's history factor and environment factors and $\alpha, \beta \in [0, 1]$.

A user's participation in communication sessions affects its absolute trustworthiness value. To help with decision-making in each session, FGN recalculates the most recent trustworthiness values of both the expected and practical session receivers. After having made decisions, FGN stores and updates the expected and practical receivers' absolute trustworthiness values in their respective personal profiles for future use.

"Trust degree", the other trust-related parameter, is fixed for each user pair in the current version of FGN design. The value can be set manually at the set-up stage of the virtual-user system.

4.5.2 Calculating Priority of Dynamic Communication Session

Session Keeper real-time calculates the priority of all suspended sessions and sends a group of sessions with highest priorities to Session Generator for regeneration. A high-priority session can either be a session with user status as emergency, a session whose status has recently changed, or a session that runs close to the end of the set-up time. In other words, user emergency status, changes in session status, and session approaching deadline all have a serious effect on session priority. The priority calculation thus needs to embody all these features. After having sent a session to Session Generator, Session Keeper deletes the session and releases involved communication resources.

Session priority (PRI) in Session Keeper determines how fast a service session can reach the point of being processed. It can be calculated from Eq. (6):

$$PRI(n) = \frac{t - t_0}{T_{LF}} \times \left\{ E_S \cdot (S_{TP} - 1) \cdot (S_{TP} - 3) \cdot (S_{TP} - 5) + \left[D_{RE} \times \left(\sum_{i=0}^{n-1} x_i + 1 \right) \right]^{S_{TP}} \right\} \tag{6}$$

n	The nth priority calculation of the session in Session Keeper.	
t	Time moment that FGN calculates the priority, $t \in (t_0,\ t_0 + T_{LF}]$	
t_0	Time moment that the session arrives at Session Keeper.	
T_{LF}	Time period that the session can live in Session Keeper.	
S_{TP}	Service type of the session. The values can be adjusted to fix into reality.	
	$S_{TP} = 10$	Emergency service.
	$S_{TP} = 5$	Real-time service, e.g., phone call.
	$S_{TP} = 3$	Real-time/Non-realtime service, e.g., messenger with voice
	$S_{TP} = 1$	Non-realtime service, e.g., email.
E_S	Emergency degree of any session involver's information, e.g., the receiver is supposed to be fire rescuing.	
	$E_S = 1000$	Constant.
D_{RE}	Demanding degree of two session involvers (i.e., the initiator and the receiver). The degree describes the closeness of the two session involvers and the value is determined by user relationship type.	
	$D_{RE} = 10$	Tense relation, e.g., boss and employee.
	$D_{RE} = 3$	Easy relation, e.g., instructor and learner.
	$D_{RE} = 1$	Relaxing relation, e.g., family members.
x_i	Value to indicate whether a user's status has changed at the i^{th} calculation.	
	$x_i = 0$	User's status has not changed since last calculation.
	$x_i = 1$	User's status has changed since last calculation.

Given that Session Keeper removes a service session when its priority value falls to zero, Eq. (6) clarifies the following facts:

- A service session only validates in Session Keeper during its connecting lifetime starting from the time it arrives at the database. (t is subject to t_0 and T_{LF}.)

- Only when the session's service type is "emergency", the session involvers' emergency status count and it counts a great deal in the calculation of priority. (E_S only operates when $S_{TP} = 10$ and E_S alone is an enormous constant number.)
- The demanding degree of the session initiator on the receiver is fixed and its absolute value has a positive effect on the calculation of priority (D_{RE}).
- The real-time feature of a session has a positive effect on the priority when the session is not urgent. The more the session requires real-time performance, the easier it is to increase the session priority. (S_{TP} positively affects the calculation of priority.)
- The constant "1" is to ensure that the service type has a positive effect on the priority calculation. ($\left[D_{RE} \times \left(\sum_{i=0}^{n-1} x_i + 1\right)\right]^{S_{TP}}$ will increase only if $D_{RE} \times \left(\sum_{i=0}^{n-1} x_i + 1\right) \geq 1$ and the more the changes in status, the more these changes may influence priority calculation.)

Eq. (6) further shows in what manner (positively/negatively or linearly/non-linearly) these parameters may affect the calculation of session priority. That is, all parameters but a session's time-to-live in Session Keeper have a positive effect on session priority. The priority value increases when the session waits longer in Session Keeper ($t \uparrow$), or related user's availability status has changed to be more reasonable for service delivery ($\sum_{i=0}^{n-1} x_i \uparrow$). The more strict the service requirement on real-time performance ($S_{TP} \uparrow$), the easier it is able to increase its priority. The increase in the demanding degree between two session involvers ($D_{RE} \uparrow$) will also improve the priority value because the closer the relationship is, the more emergency the session requires to be implemented and the faster it requires the session priority to reach the peak value. However, if a session's properties allow a long suspension time in Session Keeper ($T_{LF} \uparrow$), the priority of the session will take its time to reach the highest potential value.

4.5.3 Expressing Emergency Status of Session Involver

To show the privilege of special session involvers in session processing, we express their user emergency feature as session emergency status in FGN. The transformed session emergency feature affects the calculation of session priority in Session Keeper. FGN uses digits "0" for emergency status and "1" for non-emergency status.

4.6 Potential Implementations

To validate the virtual-user system, we can implement it either in a software environment or on a physical testbed. We will briefly introduce a testbed implementation of the system in this section and will describe in detail a software implementation in future chapter (Chapter 5).

We select an IP-Multimedia-System (IMS) testbed to implement the virtual-user system for two reasons. Firstly, IMS is a globally accepted service-delivery platform that is strongly capable of handling a variety of NGN services in theory. Secondly, the Open IMS playground at FOKUS has successfully prototyped and validated the IMS simulation environment on the testbed [85]. Based on these facts, an IMS testbed should be able to function as a suitable basement for the virtual-user system. We redraw the IMS architecture in Figure 4-16 to emphasize its key features as well as prepare an environment to import the virtual-user system.

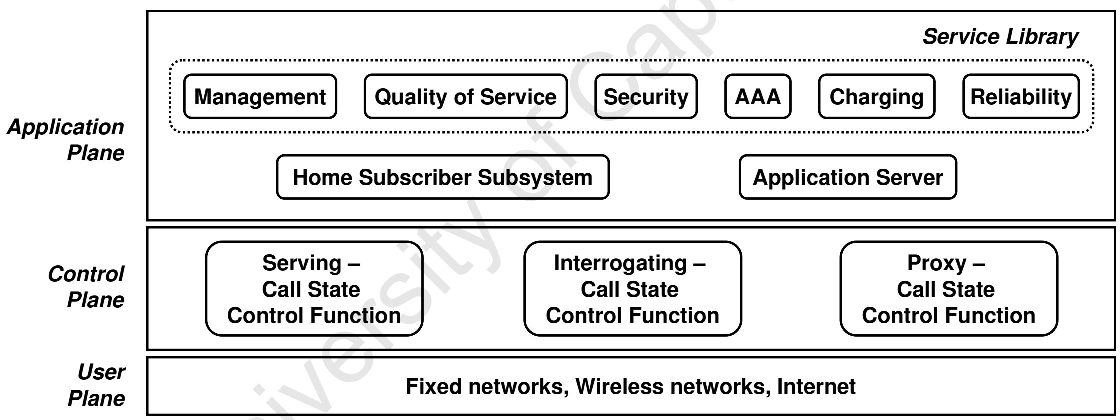


Figure 4-16. IP-Multimedia-Subsystem architecture with key components.

AAA stands for Access, Authentication, and Accounting.

In the application panel of IMS architecture in Figure 4-16, Home Subscriber Subsystem keeps all users' communication profiles such as location information, security information, and user status information. Application Server is in charge of providing existent multimedia applications and the environment to develop new applications. The service library stores specific service characteristics such as service management, quality of service, service security, and cost charging. In the control panel, Serving Call State Control Function (S-CSCF) registers a user and interacts with the application plane of the IMS architecture. Interrogating Call State Control

Function responds for the interworking between different S-CSCFs. Proxy Call State Control Function starts the operations inside the IMS.

After implementing the function modules of Figure 4-1 in the existing IMS architecture in Figure 4-16, we get an IMS system with human-like intelligence enhanced as in Figure 4-17:

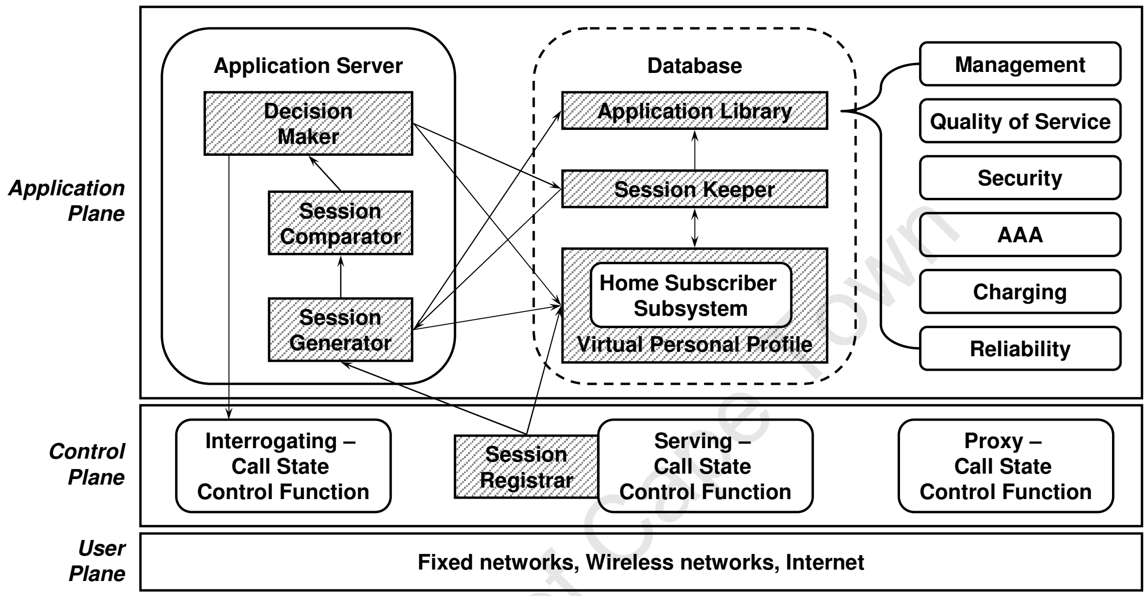


Figure 4-17. Functional implementation of human-like intelligence on IP Multimedia Subsystem.

Intelligence-related function modules are marked by light upward diagonals.

Figure 4-17 illustrates an approach of implementing the virtual-user system on the IMS platform. (1) An intelligence function module can enhance the function of an existing IMS entity, such as implementing Session Generator, Session Comparator, or Decision Maker in Application Server (AS). These operating modules of the virtual-user system each have explicit objectives of what functions to implement and are expected to conduct the functions independently. Therefore, in theory, they can be implemented as enhanced ASs that are originally designed for the development of various functions and are able to deal with dynamic content. (2) An intelligence module can also work as an extended function entity to an IMS entity, such as Virtual Personal Profile to Home Subscriber Subsystem (HSS). In addition to taking over all contents and attributes of HSS, Virtual Personal Profile especially adds in user social features. (3) An intelligence module may represent the existing IMS entities in a simple and easily accessible way, such as using an Application Library to describe all existing characteristics in the IMS service library. (4) An intelligence module can also function as an independent module on the IMS platform, such as Session Keeper and Session Registrar. Session Keeper maintains dynamically changing session information. It is therefore expected to operate as a database with specific features to handle dynamic data [82]. Session Register is a special intelligence module that probes into the control plane of the IMS platform to obtain real session information. Both S-CSCF and I-CSCF at the session initiating side can provide the information, yet I-CSCF is more suitable for accommodating Session Registar because it often contacts HSS for user connectivity. Besides, embedding Sessoin Registrar with I-CSCF makes it easier for Decision Maker to pass decision-making results about optimal connecting manner to I-CSCF through Sessoin Registrar.

Having merged intelligence modules onto the IMS platform, we use the two call-establishing procedures in Figure 4-18 and Figure 4-19 to respectively illustrate the way the intelligence enhanced IMS platform directs two-user and three-user calls (section 3.3.2).

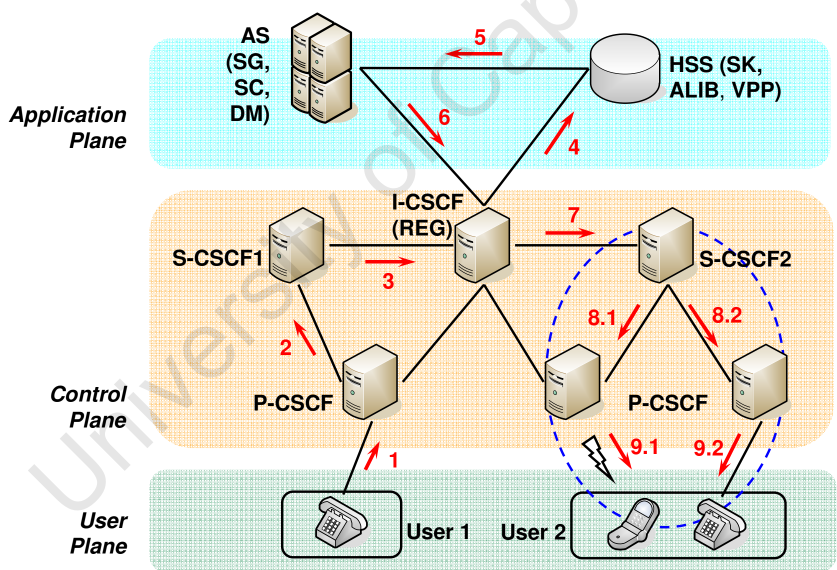


Figure 4-18. HI enhanced IMS case 1: One receiver with optional call-delivery manners.

Figure 4-18 illustrates the steps by which the human-like intelligence enhanced IMS platform delivers a phone call. When the call reaches I-CSCF where Session Registrar sits (step 3), the enhanced I-CSCF not only obtains user connectivity from HSS as usual, but also activates the intelligence procedure through Session Registar. Then the intelligence modules in the application plane conduct the intelligence selection of optimal user connectivity and send the results back to I-CSCF (steps 4, 5, and 6). In a case where the receiver has two available connecting manners (steps 8.1 and 9.1, or steps 8.2 and 9.2), only the intelligence part of the IMS platform is able to identify the better one and instruct I-CSCF to go for the theoretically optimal connecting manner.

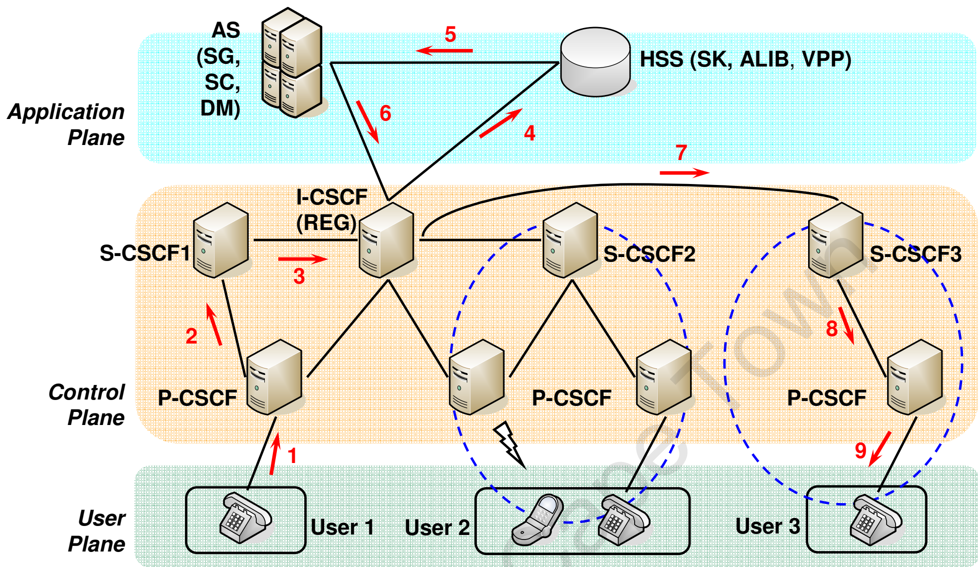


Figure 4-19. HI enhanced IMS case 2: Two optional receivers.

In Figure 4-19, the human-like intelligence enhanced IMS platform has conducted an intelligent manner-selection (steps 4 to 6) and identified a third user who is more suitable than the original user to receive the call in terms of communication availability (steps 8 and 9).

Many methods serve to test the enhanced functionality of the IMS testbed with human-like intelligence. To test its feasibility, we need to compile several experimental cases and successfully run them through the enhanced IMS testbed. To test its efficiency, we collect experimental results and provide a rational analysis. To test its ability to fulfill the initial objectives that are to better serve network users, we need to invite real users into the experiment, ask them to use the service product of the experiment, and conduct a survey to establish how satisfied they are with the service product.

However, due to limited labors and resources, the testbed implementation of the IMS system is practically impossible and thus left out. A more practical test method – software implementation – will be detailed in Chapter 5.

4.7 Summary of Chapter 4

This chapter gave a broad picture of the virtual-user system that implements human-like intelligence on the current communications network. It first explained the basic elements that traverse the system, the unique tasks of each function module, and the operating manners of each intelligence mechanism. The virtual-user system operates user-, application-, and session-elements over seven function modules: Session Registrar, Session Generator, Session Comparator, Decision Maker, Session Keeper, Virtual User Profile, and Application Library. Several mechanisms were applied to these modules to facilitate the investigation of a session in the following brief working flow. Firstly, FGN breaks a communication session and registers it with the human-intelligence part at Session Registrar, providing an opportunity for the system to detect foreseeable communication problems that are especially associated with users' availability and capability. Then Session Generator and Session Comparator cooperatively generate and analyze a pair of virtual sessions for the real session. By reviewing the analyzed results, Decision Maker decides whether to immediately deliver the session, to deliver it with less than satisfactory performance, to deliver it at a future appropriate moment, to ask for a third party's help with delivery, to suggest the expected receiver to learn from the third party, or to fail it. Thereafter, we constructed six communication scenarios to exemplify these individual and interworking mechanisms. In addition, we selected several important evaluating parameters to examine their intrinsic features and effect to the virtual-user system. Lastly, two real-world implementations of the virtual-user system were proposed.

Chapter 5 will describe the detailed experimental implementation of the virtual-user system in a Java and MySQL software environment, in terms of the basic elements, function modules, and operating mechanisms of the system.

Chapter 5 Implementing the Virtual-user System in Java/MySQL

The virtual-user system provides a technical solution for determining an optimal session-delivery manner according to user preference. On receiving an intentionally disconnected real session from the underlying NGN, Session Registrar initiates the procedure of establishing the best delivery manner for it in the human-intelligence part. Session Generator then forms several pairs of virtual sessions to propose all potential solutions for the real session according to service requirement and user status. Thereafter, Session Comparator analyzes each pair on the extent to which the practical virtual session fulfills the requirements of the original one. After having received the analysis results, Decision Maker identifies the pair that provides the most feasible session-delivery manner among all and instructs the physical network to continue the real session using the manner indicated by the identified pair. These four operating function modules obtain the user information from Virtual Personal Profile and the application information from Application Library. Session Keeper is responsible for storing suspended virtual-session pairs.

Software implementation appears to be a practical, effective, and inexpensive manner to realize such abstract and complex systems. We call the software implementation of the virtual-user system, "HIFGN project". At first, we introduce a feasible experimental environment comprising software and hardware facilities (section 5.1). Then we describe the detailed software design for the system components in terms of parameters, basic elements, and function modules (section 5.2), and the mechanisms that function in between these modules (section 5.3). Lastly, the feasible testing theories and methodologies are presented (section 5.4).

5.1 Experimental Environment

When selecting appropriate software languages to program the virtual-user system, we first need to be aware of the following issues. Firstly, the programmed system should be easily accessible, readable, and revisable. Secondly, the system should be compatible with existing platforms and testbeds to be easily merchandized in future. A bond of Java language (section 5.1.1) and MySQL language (section 5.1.2) meets the two requirements very well.

(1) These two open-source programming languages are widely available online and applicable in most fields of engineering and science. Their global accessibility brings to them a broad range of existing and potential users. Furthermore, the two languages are easily readable and rewritable for their simple syntaxes, cognitive expressions, and user-friendly interface. The openness of the two languages also ensures that the projects written in them can always receive immediate feedback from users and effective updates from experts.

(2) The majority of existing service platforms and testbeds in NGN are compatible with Java and MySQL, so is supposedly the physical-network part of FGN. For compatibility purposes, it is preferable to write the human-intelligence part of FGN (the virtual-user system) in the two languages as well. The matching of the two parts in FGN will facilitate the set-up of easy operable platforms or testbeds and the production of practical merchandise.

5.1.1 Software Facility – Java

Java ([86]) is a programming language originally developed by Sun Microsystems and released in the 1990s. It uses object-oriented programming methodology, allows programs to run on different operating systems in similar operating modes, contains built-in support for use in computer networks, securely executes code from remote sources, and has inherited the good parts of other object-oriented languages. With these wonderful features, Java technology can be easily embedded into all types of platforms and help create highly customized applications on most practical digital devices such as 3G mobile phones or remote computers. It therefore bears the brunt of the programming language for the virtual-user system.

We further choose Netbeans 5.5 ([87]), a cross-platform Java desktop, as the software-programming environment for the HIFGN project. Netbeans bundled with user-defined modules can provide users with branded customized applications. These applications are writing-once and running-anywhere due to the cross-platform feature of Java language. We can easily upgrade these applications by assembling new modules to or releasing unnecessary ones from them.

Therefore, Netbeans software tool is more than suitable for the development of the virtual-user system, which requires universal applicability in a variety of platforms or testbeds and adequate flexibility in managing its modules.

5.1.2 Software Facility – MySQL

MySQL ([88]) is an open-source relational DataBase Management System (DBMS[41] [89]) written in Structured Query Language (SQL[42] [90]). It operates multi-threads and thus is capable of serving multi-users simultaneously. MySQL language is also cross-platform available. Its MySQL server-client protocol enables users to remotely access a MySQL server from a local machine that runs MySQL client. In addition, the relational integrity constraints of the SQL language ensure that client applications cannot interfere with one another or insert inconsistent values in DBMS.

MySQL provides connectivity for client applications developed in Java via MySQL Connector/J. The MySQL Connector/J is a Java Database Connectivity (JDBC[43] [91]) 3.0 Type 4 driver that uses pure Java language, implements version 3.0 of the JDBC specifications, and communicates directly with the MySQL server using the MySQL protocol [92].

We further choose the operating tool of MySQL Query Browser ([93]) to store and retrieve data respectively in and out of the MySQL databases. We use MySQL DBMS as the user database to store dynamic user information and as the application database to store static application information in the HIFGN project.

[41] DBMS is a collection of software for organizing the information in a database concerning data input, verification, storage, and retrieval. It stores data in files and writes application-specific codes to manage it.

[42] SQL is syntax for defining and manipulating data from a relational database, developed by IBM.

[43] JDBC is a Sun-Microsystems interface standard that defines how Java applications access relational data.

5.1.3 Hardware Facility

The respective minimum hardware configurations for running Netbeans ([94]) and MySQL ([95]) in the Windows system are compared with that used in the HIFGN project, as shown in Table 5-1:

Table 5-1. Comparison of minimum- and used-hardware configurations.

	Netbeans	**MySQL**	**HIFGN Project**
Processor	500 MHz Intel Pentium III workstation or equivalent	N/A	1.7 GHz Intel Pentium M
Memory	384 MB	1 MB	1024 MB
Disk	125 MB free disk space	4 MB free disk space	1 GB free disk space
Screen resolution (for IDE[44] [96])	1024x768 pixels	N/A	1024x768
Operating system	Microsoft Windows	N/A	Windows XP

* MHz: Megahertz ($1\ MHz = 1\times10^6\ Hz = 1\times10^6\ (\sec ond)^{-1}$)

* GHz: Gigahertz ($1\ GHz = 1\times10^9\ Hz = 1\times10^9\ (\sec ond)^{-1}$)

* MB: Megabytes ($1\ MB = 1\times10^6\ Byte = 8\times10^6\ bit$)

* GB: Gigabytes ($1\ GB = 1\times10^9\ Byte = 8\times10^9\ bit$)

* N/A: Not applicable.

Comparing the details in Table 5-1 discloses that the used hardware facilities in the HIFGN project are able to meet the hardware requirements of the two software facilities.

5.2 Software Design – Parameters, Elements, and Modules

To establish the HIFGN project, the first and necessary task is to explicitly design the parameters, elements, modules, and mechanisms of the virtual-user system. This section focuses on the first three and the next section on the last. The second task is to practically program these

[44] Integrated development environment (IDE) is a programming environment packaged as an application program, typically consisting of a code editor, a compiler, a debugger, and a graphical-user-interface builder. It can be a stand-alone application or part of one or more existing and compatible applications, such as Sun IDE.

system components in Java and MySQL languages. The last is to validate the programmed system (Chapter 6) and analyze the system performance (Chapter 7).

5.2.1 System Parameters

System Timer is an essential system parameter that sets a time regulator for all components of the virtual-user system. (1) A running System Timer represents the real-time availability of the time-sensitive user- and session-elements. Each of these elements carries a time stamp that indicates session set-up time. By referring to the session set-up time and the current system time, we are able to calculate the collapsed period since session set-up. (2) System Timer synchronizes system function modules so that the completion of a task in one module can activate the execution of a task in another. The latter needs to real-time listen to the former on its most recent operating status and actions. (3) System Timer makes it possible for some function modules such as Session Keeper to perform tasks at successive moments.

System Timer is mainly regulated in the System-Time class in the "hifgn.system" package. Each System-Timer instance contains the information of year, month, day, hour, minute, and second at the current moment. It is able to display any combination of these characters.

5.2.2 Information Databases

Information databases store and organize the relevant information for intelligently processing communication sessions in the human-intelligence part. They include two relational databases: the user- and application-databases. The two databases adopt central management in monitoring the content of information for consistency and distributed management in the form and location of storage for easy accessibility.

The user database keeps the communication profiles of all network users and structures the data in two types of table. (1) One type (Table B-2 to Table B-5) stores user information that includes personal details, social relations, and current communication status in respective records. The number of the records changes with a user's joining or leaving the network and these tables are therefore unfixed in size. (2) The other type (Table B-6 to Table B-9) maintains the references for intelligence-related human characteristics. For example, a user's social-relationship table maintains the fields of relationship ID, relationship-affiliated social domain, demanding degree, relationship type of the counter-part, and its corresponding ID. These tables statically store human nature in the form of network-recognizable characteristics.

The application database keeps the specifications of the applications adopted by FGN and structures the data in two types of tables as well. (1) One type (Table B-10) indexes all the applications and lists their individual intrinsic characteristics. Any function module can refer to these tables for needed application-performance information to help with evidential evaluation or reasonable decision-making. (2) The other type (Table B-11) only includes one table that enumerates the available devices for each application. By comparing the devices required by an application (obtained from Table B-10 “application_index”) with those available for a user (obtained from Table B-8 “activity_2_location” and Table B-3 “personal_detail” in the user database), the system is able to determine whether the user is available for the application.

Figure 5-1 illustrates the software design of the information databases in the project.

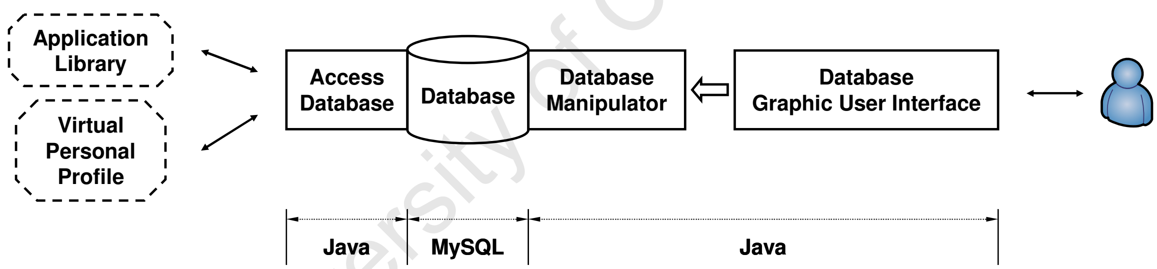


Figure 5-1. Software design of the information databases.

From Figure 5-1 to Figure 5-7, an octagon stands for a function module in Java programming language, a rectangle for a major class in the module, and a cylinder for a database in MySQL programming language. The components surrounded by solid lines are the ones accented in the section where they are depicted and those surrounded by dashed lines are the ones not accented but relevantly connected to the former. Wide arrows indicate the interconnection between the classes inside a module and thin arrows between the modules.

In Figure 5-1, the essence of the information databases – Database – is manipulated in the MySQL tool. The modules Application Library and Virtual Personal Profile fetch information from Database through the access entry such as the Access-Database class. In addition, users are able to customize their communication information and service providers are able to update the application information using the Database-Manipulator class through the user-friendly Database-Graphic-User-Interface class. The last three classes are written in Java language.

5.2.3 Basic Elements

The HIFGN project manipulates three basic elements, including user element, application element, and session element. These elements are programmed in Java language.

5.2.3.1 User Element

A user element describes a real user's communication profile, including personal details, social relations, and current communication status. In the HIFGN project, we represent a user's current communication status as a combination of his/her preset communication schedules with the most recently updated status from his/her last communication event. The user element is mainly regulated in the Element-User class in the package "hifgn.element.user". Table 5-2 lists the major characteristics of the class.

Table 5-2. Characteristics of the User-Element class.

Variable	Type	Explanation	Notes
user_id	String	ID to uniquely define a user	IPv6 address[45] ([97])
user_name	String	User name	Maximum 64 characters
trustworthiness	Integer	User's absolute trustworthiness	[0, 100]
emergency	Integer	User's emergency status	"0" for emergency "1" for non-emergency
location	Vector	User's default communication locations	
device	Vector	Available devices at each location	Obtained from *location*
schedule	Vector	User's weekly schedule	
activity	Vector	Social activity in each scheduled timeslot	Obtained from *schedule*
relation	Vector	User's socially related users	
timestamp	SystemTime[Δ]	Time that the element sets up	
session_ids	Vector	Currently involved sessions' IDs	IPv6 address

* Vector: In Java programming language, the Vector class implements an expandable array of objects that are accessible by an integer index. The size of a Vector can grow or shrink as needed to accommodate adding or removing objects [98].

[Δ] SystemTime: A created Java class that marks the element-setup time (section 5.2.1).

[45] Internet Protocol version 6 (IPv6) is a network layer protocol for packet-switched data networks. It supports 2^{128} addresses, or approximately 5×10^{28} addresses for each of the roughly 6.5 billion people alive today.

Noticeably, for a user involved in an ongoing session in the human-intelligence part, the system uses a set of user elements to describe his/her communication statuses at different moments. These elements share the same content of the user's fixed information such as user ID and differ in the content of the user's time-sensitive information such as current activity.

5.2.3.2 Application Element

An application element is a piece of information that embodies an application by giving a correct level of value for each application characteristic. It is mainly regulated in the Element-Application class in the "hifgn.element.application" package, as shown in Table 5-3.

Table 5-3. Characteristics of the Application-Element class.

	Variable	Type	Explanation	Notes
√	*application_id*	String	ID to uniquely define an application	IPv6 address
√	*application _name*	String	Application name in the HIFGN project	Maximum 32 characters
√	*service_type*	String	Type of service that the application carries	
√	*media_type*	String	Media type of the service	Data, audio, or video
√	*connecting_time*	Integer	Period taken to set up the application	
√	*bandwidth*	Bandwidth$^{\Delta}$	Required bandwidth for the application	
	packet_delay	PacketDelay$^{\Delta}$	Allowed packet-delay for the application	
	bit_error_rate	BitErrorRate$^{\Delta}$	Allowed bit-error-rate for the application	
√	*availability*	UtilAvailability$^{\Delta}$	Availability of the application	
√	*devices*	UtilDevice$^{\Delta}$	Devices available for the application	
√	*billings*	Vector	Billing types to the application	

$^{\Delta}$ Bandwidth: A created class that describes a bandwidth range, comprising low bound, high bound, and unit.

$^{\Delta}$ PacketDelay: A created class that describes a range of packet-delay, comprising minimum- and maximum-allowed delays.

$^{\Delta}$ BitErrorRate: A created class that describes a range of bit-error-rate (ber), comprising lowest- and highest- ber.

$^{\Delta}$ UtilAvailability: A created class that describes the immediate availability of an application.

$^{\Delta}$ UtilDevice: A created class that enumerates all used types of devices in the project. The content is the same as "stringDevice" in Table B-2.

* Characteristics marked by " √ " are used in the coding for the HIFGN project.

For the purposes of simplicity and easy simulation, we have only selected a few major characteristics of an application in the HIFGN project and will further explain the detailed levels of value used for each characteristic (section 5.3.8).

5.2.3.3 Session Element

A session element is a piece of information that embodies an ongoing session in the human-intelligence part. It is composed of the essence of an application and several session involvers. The session element is mainly regulated in the Element-Session class in the "hifgn.element.session" package. Table 5-4 lists the major characteristics of the class.

Table 5-4. Characteristics of the Session-Element class.

Variable	Type	Explanation	Notes
session_id	String	ID to uniquely define a session	IPv6 address
application	ElementApplication $^{\Delta}$	Session carried application	
initiator	ElementUser $^{\Delta}$	Involved session initiator	
receiver	ElementUser	Involved session receiver	
thirdparty	ElementUser	Involved session third-party	
t_setup	SystemTime	Moment that the session sets up	
t_lasted_refresh	SystemTime	Moment that the session is requested	
priority	Double	Priority value of the session in the module Session Keeper (SK)	Special for SK
num_of_changes_in_status	Integer	Number of changes in session involvers' communication status when the session is alive in SK	Special for SK
emergency_status	Integer	Emergency status of session involvers	Special for SK

$^{\Delta}$ ElementApplication: A created Java class for application elements (section 5.2.3.2).

$^{\Delta}$ ElementUser: A created Java class for user elements (section 5.2.3.1).

* The characteristics in the solid-line-surrounded cells are applicable to any related function module whereas those in the dashed-line-surrounded cells are especially for the module Session Keeper.

5.2.4 Function Modules

We implement all seven function modules of the virtual-user system in the HIFGN project, including Session Registrar, Session Comparator, Session Generator, Decision Maker, Session Keeper, Virtual Personal Profile, and Application Library. In addition, we include Session Simulator in the project for the purpose of testing the programmed system (section 5.4.1).

5.2.4.1 Session Registrar

The function module Session Registrar registers a real communication session in the human-intelligence part by intercepting the essential information of the session and further filling it in a set of session-element containers, as shown in Table 5-5. It is regulated in the package "hifgn.module.reg", as shown in Figure 5-2.

Table 5-5. Interpreted real-session information in the Session-Registrar module.

Variable	Variables Description	Notes
session_id	Representing a session	The system randomly chooses an available ID from a database that contains all session IDs
application_id	Application of the session	For generating application elements in Session Generator
initiator_id	Session initiator	For generating related user elements in Session Generator
calling_device		For updating user information in Virtual Personal Profile
receiver_id	Session receiver	For generating related user elements in Session Generator
t_setup	Set-up time of the session	An instance of the System-Time class

Once having interpreted the key information (Table 5-5), the Session-Registrar module proposes sets of session-, application-, and user-elements to represent the real session. By fairly understanding these element sets in terms of format and content, the rest of the human-intelligence part is then able to establish an intelligence approach for session delivery.

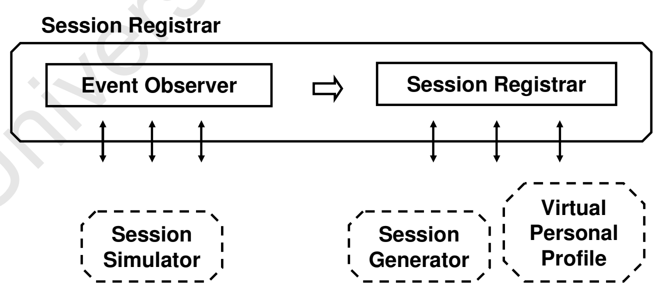


Figure 5-2. Layout of the classes in the "hifgn.module.reg" package.

The Event-Observer class sends input events to Session Simulator. The Session-Register class registers sessions in Session Generator and communicates with Virtual Personal Profile for necessary information.

5.2.4.2 Session Generator and Session Comparator

The function module Session Generator generates several pairs of virtual sessions for a real session and the module Session Comparator compares the virtual-session pairs, supplying the results for further analysis. Because of their tight correlation, these two modules are regulated together in the package “hifgn.module.sgc”, as shown in Figure 5-3.

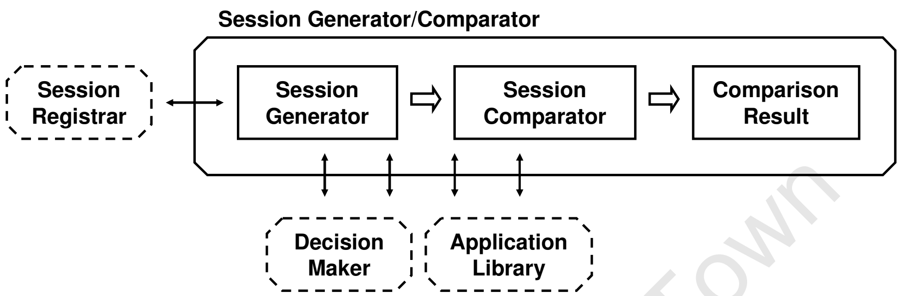


Figure 5-3. Layout of the classes in the “hifgn.module.sgc” package.

The Session-Generator class and the Session-Comparator class are respectively responsible for generating and comparing a pair of virtual sessions. These classes only perform actions when receiving a request from other classes and they do not keep records of the sessions. The Comparison-Result class organizes the comparison results in a fixed format that is understood by the module Decision Maker as well.

The Session-Generator class talks to Session Registrar and both the Session-Generator and the Session-Comparator classes talk to Decision Maker and Application Library.

5.2.4.3 Decision Maker

On getting the comparison results, the function module Decision Maker makes decisions on whether the created virtual session is able to meet the requirement of the expected one and, if yes, to what extent. It is regulated in the package “hifgn.module.dm”, as shown in Figure 5-4.

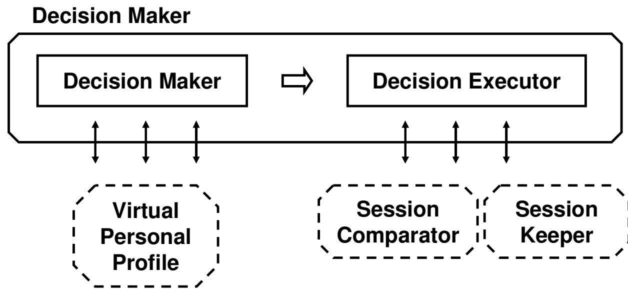


Figure 5-4. Layout of the classes in the "hifgn.module.dm" package.

The Decision-Maker class makes decisions on how to optimally deliver a session by considering both the comparison results from Session Comparator and the session involvers' relationship from Virtual Personal Profile. After having worked out the final delivery manner, the Decision-Executor class demands that the related modules carry out the decision and coordinates them during session-delivery execution.

The Decision-Maker class makes decisions by referring to Virtual Personal Profile and the Decision-Executor class receives comparison results from Session Comparator and stores temporarily suspended sessions in Session Keeper.

5.2.4.4 Session Keeper

The function module Session Keeper manages the suspended virtual sessions. It is regulated in the package "hifgn.module.sk" (SK), as shown in Figure 5-5.

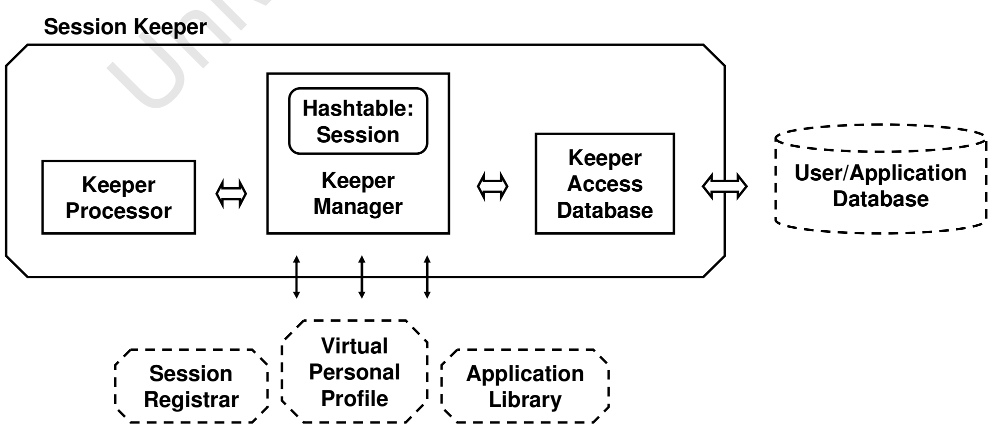


Figure 5-5. Layout of the classes in the "hifgn.module.sk" package.

In most cases, more than one session is suspended in Session Keeper at the same time. Therefore, Session Keeper uses a hashtable[46] ([99]) to store these sessions and especially sets up a Session-Manager class to manage the hashtable. This management includes adding, deleting, obtaining, and modifying a session. The Session-Processor class mainly performs three tasks that include calculating the session priority, recording the number of changes in the receiver status, and enquiring of the session involvers' emergency status. The last two assist with the calculation.

The Session-Processor class obtains the changes in the receiver's status by comparing the status newly received from Virtual Personal Profile with that known to Decision Maker. To obtain the newest user status, the class takes the initiative to regularly ask for user information and is always ready to receive an update from Virtual Personal Profile. The parameter "*num_of_changes_in_status*" increases with the number of changes in the receiver's status. For example, if Lisa's communication status is office-phone-available at 4pm, cell-phone-available at 5pm, and home-phone-available at 6pm, the number of changes in her communication status during those three hours is "two".

Furthermore, the status of a session is defined "emergency" when any of the session initiator, receiver, or third party has an "emergency" feature. The parameter "*emergency_status*" in session elements thus depends on user status. It is initially set to reflect the emergency status of the original session initiator and receiver. Later on, the *emergency_status* parameter may change if the session brings in a third party. Using number "0" for an emergency status and number "1" for a non-emergency status, the final emergency status can be determined as shown in Table 5-6:

Table 5-6. Rules for obtaining final emergency status.

Existing emergency status	New emergency status	Final emergency status
0	0	0

[46] Hashtable is a data structure that uses a key (e.g., a person's user ID in network) to search for the corresponding value (e.g., that person's user name). It works by transforming the non-contiguous key into a hash number and using the number as an array index to locate the desired value in the array.

0	1	0
1	0	0
1	1	1

Table 5-6 shows that, if either the existing emergency status or the new one is "emergency", the final status is "emergency".

The Keeper-Manager class reads information from Virtual Personal Profile and Application Library when necessary and writes session-processing results into Session Generator.

5.2.4.5 Virtual Personal Profile

The function module Virtual Personal Profile stores the communication profiles of the users who are involved in the ongoing sessions in the human-intelligence part. It is regulated in the package "hifgn.module.vpp", as shown in Figure 5-6.

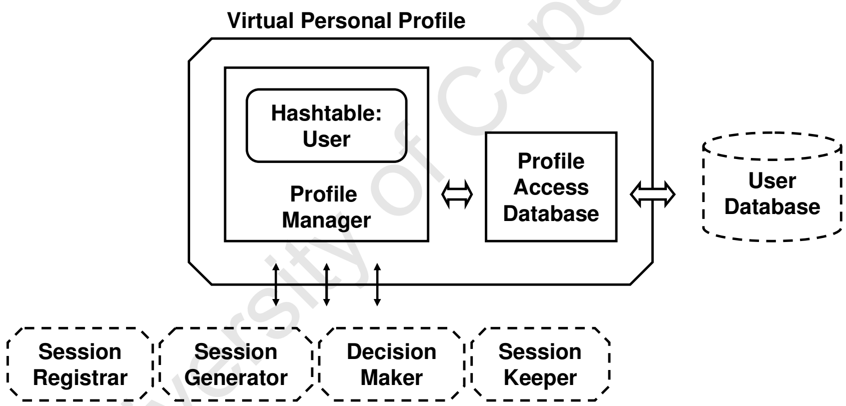


Figure 5-6. Layout of the classes in the "hifgn.module.vpp" package.

The Profile-Manager class manages a hashtable that accommodates all the users involved in the ongoing sessions in the human-intelligence part. This management mainly includes adding, deleting, obtaining, and modifying a user element. When adding a new user to the hashtable, the class collects the user information from the user database via the Profile-Access-Database class.

The Profile-Manager class provides an interface for Session Registrar, Session Generator, Decision Maker, and Session Keeper to obtain information from the user database.

5.2.4.6 Application Library

The function module Application Library stores the specifications of applications that are selected in the HIFGN project to represent the prevalent ones in the current communications network. It is regulated in the package "hifgn.module.alib", as shown in Figure 5-7.

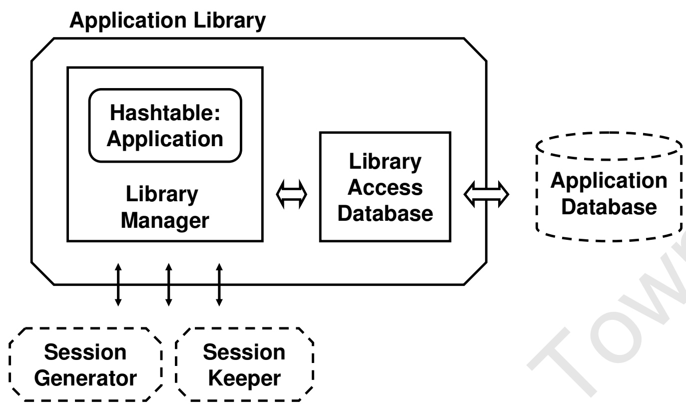


Figure 5-7. Layout of the classes in the "hifgn.module.alib" package.

The Library-Manager class manages a list of applications in a hashtable so that other modules can read the necessary information systematically and fast. This management includes adding, deleting, obtaining, and modifying an application. To insert new applications or modify existing ones, Application Library reads via the Library-Access-Database class the information from the application database, where application information is frequently updated by other network resources such as service providers.

The Library-Manager class provides an interface for Session Generator and Session Keeper to obtain information from the application database.

5.3 Software Design – Mechanisms

Many mechanisms enhanced with properly chosen parameters operate over or interconnect between function modules, resultantly contributing to the implementation of intelligence. Below are the detailed designs of the major mechanisms.

5.3.1 Locating Practically Available Devices for a User at an Instant

The search procedure implied in Figure 5-8 exhibits the way that the system identifies the most likely available devices for a user at a given time. This procedure applies to user elements.

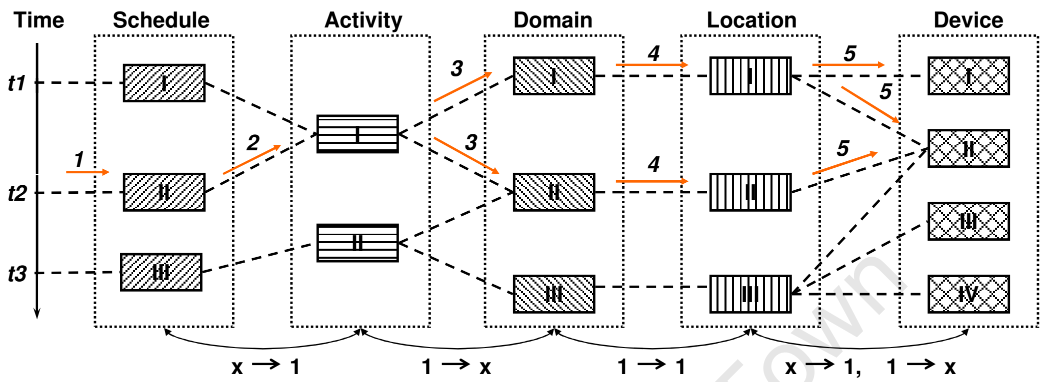


Figure 5-8. Procedure of determining current-time-available devices for a user.

The chain of determining available devices for a user sequentially goes through six issues, with the time issue shown as a down arrow and the rest as five dotted rectangles with each containing several example elements as marked rectangles. The elements belonging to one issue relate to those belonging to another issue and the relations are indicated by dashed lines. Mark "x->1" means that several elements in the left issue can have a deal with one element in the right issue. Mark "1->x" means that one element in the left issue relates to several elements in the right issue. Mark "1->1" means one element in the left issue only relates to another one in the right issue.

A procedure moving through Figure 5-8 from left to right is able to identify the available devices for a user at a given time. A series natural numbers in an increasing sequence gives an example of such a procedure. On receiving the request for a user's available devices at a time moment ($t2$), the system searches the user's schedule (Schedule II) and, based on the results, locks to the user's activity (Activity I) at that moment. This activity might happen in several social domains (Domain-I and II) with each involving several physical locations (respectively Location-I and II). Each location determines a number of available devices (Device-I and II).

A social domain implies metaphorical fields where a group of social individuals share characteristics or knowledge, have common interests, performs similar communication tasks, and influence each other's communication status in a certain way. It has general requirements on the real-time availability, performance, and billing manner of communication services and has nothing to do with a specific moment. A physical location refers to a geographic position where a communication event occurs. It relates closely to a user's communication schedule at that moment. In reality, a social domain involves several physical locations. For example, a karate-club domain may accommodate two physical locations: the training field and the competition field. The project uses one location to represent a domain for simplicity purposes.

5.3.2 Approaching Potentially Expected Devices for an Application

The search procedure implied in Figure 5-9 exhibits the way that the system identifies the available devices for an application. This procedure applies to application elements.

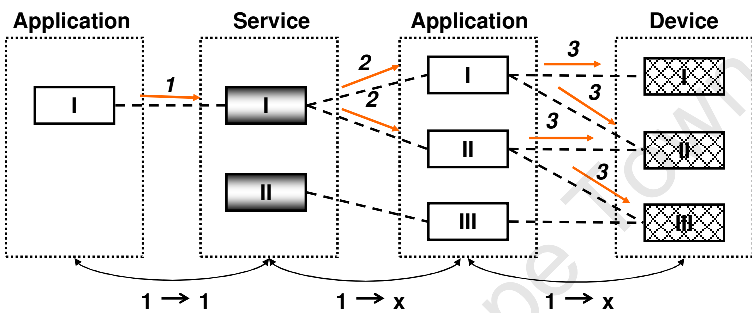


Figure 5-9. Procedure of using service type to determine available devices for an application.

A procedure moving through Figure 5-9 from left to right extends the range of available devices for an application by involving other potential applications, which serve the same function in a session. A series natural numbers in an increasing sequence gives an example of such a procedure. Given the expected application (Application I), the system first discovers the service that the application carries (Service I) and then identifies all the applications that are able to perform the function of the service (Application I itself and Application II). Because each application has its own available devices, the number of potential devices for the session increases with the involvement of new potential applications. That is, the originally expected application (Application I) has two usable devices (Device-I and II) and, therefore, initially only two devices are available for the session. However, after the system has found an alternative (Application II) that brings in a new usable device (Device III), three devices are currently available for the session (Device-I, II, and III).

5.3.3 Comparing a Pair of Virtual Sessions

To compare the created virtual session with the expected one, the system identifies the fulfilment of the former on the deliverability requirements of the latter. The identification process in the HIFGN project specifically uses the user-based session-delivery approach (section 4.2.3). The successful delivery of a session needs to first meet the service requirements of the original initiator on the original receiver, ensuring their service performance as much as possible, and then consider using other human resources. Therefore, the service performance and the strictness of users' social relationship count the most during the identifying process.

Firstly, we select two sets of parameters to represent the service performance: real-time availability and quality of service. The first represents the urgency degree to which a session requires a delivery. It determines the length of the period allowed for a session to consider using other human resources. The second evaluates service performance in all aspects such as bandwidth, packet delay, bit error rate, or billing. For two applications carrying the same service, if one is able to provide broader bandwidth, shorter packet delay, or lower bit error rate than the other, the former is said to be able to provide a better quality of service.

Secondly, we introduce a scheme to map the strictness of six common social relationships to the categories of requirement on session delivery in terms of the above two sets of parameter in Table 5-7.

Table 5-7. Effect of social-relationship strictness on session delivery in various domains.

Social Domain	Immediately Available			Available In future			Strictness of Social Relationship
	Required Quality	Better Quality	Poorer Quality	Required Quality	Better Quality	Poorer Quality	
Business	1	0	0	0	0	0	Strict
Family	1	1	1	1	1	0	Easy
Friend	1	1	0	1	1	0	Performance
Club	1	1	1	0	0	0	Real-time
Other	1	0	0	0	0	0	Strict
Unknown	0	0	0	0	0	0	No-relation

* The "Social Domain" column lists six types of social domains, with each determining a type of social relationship. In the middle six columns, if session involvers have a relationship indicated by the cell value in the "Social Domain" column in the same row, cell value marked "1" means that the session allows a combination of requirements described in the heading and that marked "0" means that the session does not allow such a combination. According to the relationship of the involvers and the corresponding requirements, the "Strictness of Social Relationship" column concludes the type of strictness that the session involvers in different domains respectively require on their sessions.

In Table 5-7, we categorize the users' requirements on the urgency and performance of session delivery according to their social relationships. (1) Users in the "Business" domain have "Strict" requirements on both the real-time availability and the quality of service for their sessions. (2) Users in the "Family" domain have such an "Easy" requirement on session delivery that they are only concerned about whether a session is deliverable or not. They have no specific requirement on any feature and thus can tolerate a session with any level of availability and quality. (3) Users in the "Friend" domain allow certain time delay in session processing but have a minimum requirement on service "Performance". (4) Users in the "Club" domain have "Real-time" availability requirement but no strict quality requirement on session delivery. (5) Users in the "Other" domain have "Strict" requirements on real-time availability and quality of service. (6) Users in the "Unknown" domain have an undetermined relationship and, if the system has successfully identified a relationship of them, the strictness requirement should be "Strict".

Referring to the two session-delivery affecting factors – the service performance and the strictness of user social relationship, we further set up a set of logic rules in Figure 5-10 to illustrate the procedure of determining a proper type of session delivery.

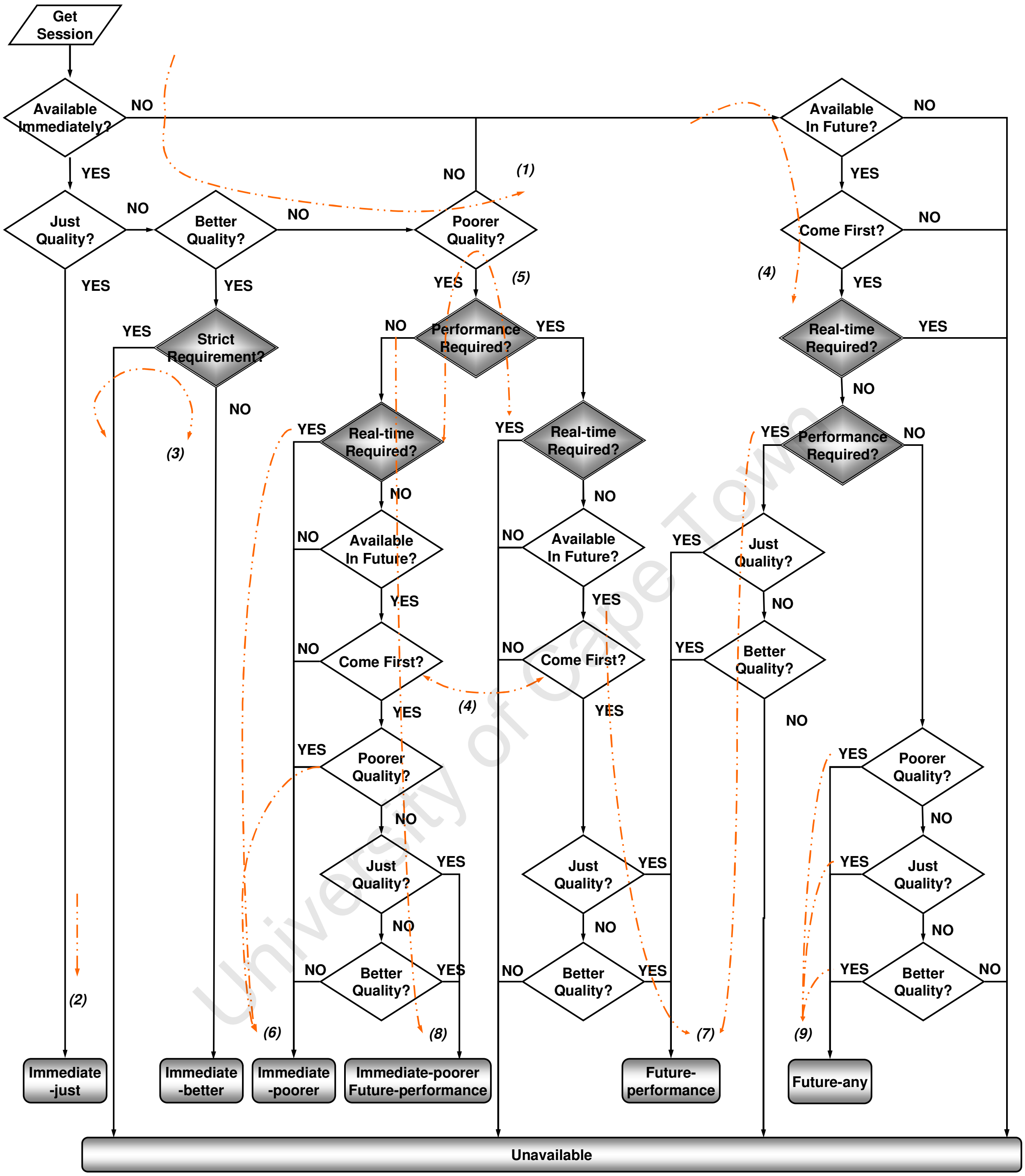


Figure 5-10. A scheme to determine the deliverability of a session.

The judging conditions embraced in the plain diamonds derive from service performance and those embraced in the diamonds shaded with variants from centre derive from user social relationships. The rounded rectangles shaded with horizontal variants contain the comparison results. “Future-performance” stands for either “future-just” or “future-better”. “Future-any” stands for “future-just”, “future-better”, or “future-poorer”.

There are many flexible schemes to identify whether a created virtual session is deliverable or not, each to serve the users with a different requirement on session delivery. The information conveyed in Figure 5-10 illustrates one of them. (1) The system first checks the virtual session's real-time availability and, if available, then checks its quality of service. If several potential sessions are available at the checking moment, the ones providing the required-, better-, and poorer-quality individually have the highest-, middle-, and lowest-priority in terms of delivery. (2) If the session fairly meets the requirements on both real-time availability and quality of service, the session is to be successfully delivered. (3) If the session users have a strict requirement on session delivery, the sessions that provide better quality of service will be failed. (4) If some sessions are unavailable at the checking moment but allow a delay in their connecting set-up stage, the system will then check their future availability. In such a scenario, whichever session becomes first available has the highest priority, because the system attempts to deliver a session as soon as possible to satisfy the users in the first place and avoid causing extra traffic in future communications. (5) In general, a session that is immediately available has precedence over the one available in future when they provide the same type of quality. However, a future available session with required- or better-quality is able to compete with an immediately available one with poorer quality. Using either of the two sessions depends on the users' preference. (6) For the users who only request real-time session delivery such as those in "Club" domain, the requirement on real-time availability precedes that on quality of service. (7) For the users who are only concerned of service performance such as those in "Friend" domain, the requirement on quality of service precedes that on real-time availability. (8) Some users such as those in "Family" domain are very flexible in using either immediate-available poorer-quality service or future-available quality-guaranteed service. (9) In a reserved scenario, the users have the right to choose any level of quality for their sessions after a session delay.

The above scheme produces seven types of results as listed in Table 5-8. Each result exhibits whether a newly created virtual session is able to meet the requirements of the expected one and, if yes, concerning which aspects. These results together prepare for decision-making on how to deliver a session in a reasonable way.

Table 5-8. Results obtained after comparing availability and service-quality for a session.

Available Immediately			Available In future			Unavailable	Result
Required Quality	Better Quality	Poorer Quality	Required Quality	Better Quality	Poorer Quality		
0	0	0	0	0	0	1	Unavailable
1	0, 1	0, 1	0, 1	0, 1	0, 1	0	Immediate-just
0	1	0, 1	0, 1	0, 1	0, 1	0	Immediate-better
0	0	1	0	0	0, 1	0	Immediate-poorer
0	0	1	1	0, 1	0, 1	0	Immediate-poorer Future-performance
0	0	1	0, 1	1	0, 1	0	
0	0	0	1	0, 1	0, 1	0	Future-performance
0	0	0	0, 1	1	0, 1	0	
0	0	0	1	0, 1	0, 1	0	Future-any
0	0	0	0, 1	1	0, 1	0	
0	0	0	0, 1	0, 1	1	0	

* In the left six columns, cell value "1" means that the session is able to meet the combined requirements described in the heading whereas "0" means not. "0, 1" means that it does not matter to the session whether the requirements are met or not. Values in the "Result" column number the result types. Adjacent rows with the same result represent different scenarios where a session gets the same treatment.

For a virtual-session pair in the HIFGN project, we use the available devices for the session receiver to represent the created virtual session and use the application-required devices to represent the expected virtual session. Therefore, checking the deliverability of the created virtual session is to check whether the devices available for the receiver (short as "Devices", section 5.3.1) are the ones that meet the original requirements by the initiator ("First Device"), the ones with better quality than required ("Better Device"), or the ones with poorer quality than required ("Worse Device"). We obtain the latter two types of session by identifying potentially available applications (section 5.3.2).

With respect to the device availability, we can explain the results of Table 5-8 in this way. (1) Result "Unavailable" indicates the scenario where no available devices are able to meet the requirements now and in the future. (2) Result "Immediate-just" indicates the scenario where the First Device is among the immediately available Devices. (3) Result "Immediate-better" indicates the scenario where the Better Device is among the immediately available Devices whereas the First Device is not. (4) Result "Immediate-poorer" indicates the scenario where only the Worse Device is among the immediately available Devices. (5) Result "Immediate-poorer and future-performance" indicates the scenario where the Worse Device is among the immediately available Devices and either or both of the First Device and the Better Device are among the Devices available in future. (6) Result "Future-performance" indicates the scenario where no devices are immediately available whereas either or both of the First Device and the Better Device are among the Devices available in future. (7) Result "Future-any" indicates the scenario where no devices are immediately available whereas at least one of the First Device, the Better Device, and the Worse Device are among the Devices available in future.

5.3.4 Making Decision According to Service Performance and Social Relation

In the HIFGN project, we try best to use the originally preferred human resources and only consider involving new ones when necessary, as is the so-called user-based session-delivery approach (section 4.2.3). Guided by the manner, the system is able to successfully determine an optimal delivery manner for a session. The information in Table 5-9 exemplifies such determining process by using the virtual-session comparison results in Table 5-8.

Table 5-9. Determining session delivery according to comparison results of session pair.

Priority	Manner of	Original Receiver	Strictness requirement on:		Assistant Receiver	Strictness requirement on:	
			Real-time	Performance		Real-time	Performance
1	*deliver*	√	√	√			
2	*force*	√	√				
2	*postpone*	√		√			
3	*help*				√	√	√
4	*learn*				√		√
5	*fail*						

* Sequential numbers from "1" to "5" indicate the levels of priority in a descending order.

* Mark " √ " indicates that the comparison result is able to meet the requirements described in the heading.

In Table 5-9, the system identifies an optimal manner to deliver a virtual session based on its comparison results with the expected one using the user-based session-delivery approach. (1) If a virtual session is able to provide an immediately available quality-guaranteed device for the originally expected receiver, it then possesses the highest priority and the system immediately delivers the session. (2) If a virtual session is able to provide either an immediately available poorer-quality device or a future available quality-guaranteed device to the originally expected receiver, it possesses the second highest priority. Based on the session involvers' social relationship, the system either forces or postpones the session to the originally expected receiver. (3) If a virtual session cannot provide any type of available devices for the originally expected receiver but it can find another receiver who has quality-guaranteed devices immediately available, the system then asks the assistant receiver to help with the session. (4) If a virtual session cannot directly find any available device through any potential user, it will learn what to do according to user social relationships. (5) Lastly, if the system cannot work out a proper session-delivery manner after having tried all the above ones, it will instruct to fail the session.

Furthermore, if more than one created virtual session is practically available for a receiver, they need to compete for the final selected session. Table 5-10 lists out a detailed mapping from the service performance to the final determined delivery manner under a specific user social relationship for two such virtual sessions.

Table 5-10. Identifying the delivery manner for two created sessions with the same receiver.

	S 1'	**S 2'**		**S 3'**		**S 4'**		**S 5'**	
S 1	S1 or S1'	S 1		S 1		S 1		S 1	
S 2	-	*Strict*	fail	*Strict*	fail	*Strict*	fail	*Strict*	fail
		Real-time	S2 or S2'	*Real-time*	S 2	*Real-time*	S 2	*Real-time*	S 2
		Performance	S2 or S2'	*Performance*	S 2	*Performance*	S 2	*Performance*	S 2
		Easy	S2 or S2'	*Easy*	S 2	*Easy*	S 2	*Easy*	S 2
		No-relation	learn	*No-relation*	learn	*No-relation*	learn	*No-relation*	learn
S 3	-	-		*Strict*	fail	*Strict*	fail	*Strict*	fail
				Real-time	S3 or S3'	*Real-time*	S 3	*Real-time*	S 3
				Performance	help	*Performance*	*S 4'*	*Performance*	help
				Easy	S3 or S3'	*Easy*	*S 4'* or S3	*Easy*	S 3
				No-relation	learn	*No-relation*	learn	*No-relation*	learn
S 4	-	-		-		*Strict*	fail	*Strict*	fail
						Real-time	help	*Real-time*	help
						Performance	*S4 or S4'*	*Performance*	*S 4*
						Easy	*S4 or S4'*	*Easy*	*S4 or S5'*

					help		help
				No-relation	learn	*No-relation*	learn
S 5	-	-	-	-		*Strict*	fail
						Real-time	help
						Performance	help
						Easy	<u>*S5 or S5'*</u>
						No-relation	learn

* S 1 / S 1': Session 1 / Session 1' for immediately available session with just quality of service.

* S 2 / S 2': Session 2 / Session 2' for immediately available session with better quality of service.

* S 3 / S 3': Session 3 / Session 3' for immediately available session with poorer quality of service.

* S 4 / S 4': Session 4 / Session 4' for available-in-future session with just or better quality of service.

* S 5 / S 5': Session 5 / Session 5' for available-in-future session with poorer quality of service.

** Strict: The relationship of the initiator and the receiver requires the newly created session to be immediately available and be able to provide just or better quality of service.

** Real-time: The relationship of the initiator and the receiver requires the newly created session to be immediately available, regardless of quality of service.

** Performance: The relationship of the initiator and the receiver requires the newly created session to be able to provide just or better quality of service, regardless the time of becoming available.

** Easy: The relationship of the initiator and the receiver allows the flexibility in the newly created session in terms of real-time availability and quality of service.

** No-relation: (same as the above description of "Strict".)

*** Future-available sessions are expressed in italic, i.e., *S 4'*.

*** Optional sessions are underlined, i.e., <u>*S 4 or S 5'*</u>, and they abide by first-available-first-use principle.

The information conveyed in Table 5-10 discloses the rules of identifying a system-preferred virtual session out of the two compared and determining the corresponding delivery manner to the identified one. We list the five types of created session, with each undertaking a type of comparison results, individually as row names and column names. Each cell value comprises the final identified session and its corresponding delivery manner under a specific user social relationship. We then apply the following rules to assist with the identification. (1) By adopting the user-based session-delivery approach, the system attempts to meet the requirements of the original receiver on service performance as much as possible. (2) If the two compared sessions such as S3 and S3' have the same comparison results, the system uses the one that becomes available first. (3) Only when the final determined session is S1 or S1', the manner of delivery is *deliver*; for the rest, the manner of final delivery is *force*. (4) Due to the user-centric feature, the *force* manner has a higher priority than those of the *help* and *learn* manners. (5)

Session priority, connecting lifetime, and session ID cooperatively contribute to the determination of a final delivery manner. The eventually identified session will be the one with higher priority, with shorter connecting time when the priorities are the same, or with smaller ID when the other two factors are the same. (6) A suspended session such as S4/S4' or S5/S5' only validates within its connecting lifetime. (7) The system needs referring to additional conditions for a final decision-making in some scenarios, such as that when comparing the sessions S4 and S4' with the involvers being in an "Easy" relationship. If S4, S4', or both become available in a short time, the system will wait for the one that first becomes available. Whereas, if both S4 and S4' become available after a long time, the system then asks the third party for help.

We use the scenario of comparing the sessions S3 and S4' as an example to illustrate the determination process, referring to the shaded block in Table 5-10. Of the two compared sessions, the S3 is immediately available but with poorer quality of service and the S4' is only available in future expected with just/better quality. If the sessions involvers start the session in a business domain where a "Strict" requirement applies to the sessions occurring there, the system has to fail both the S3 and the S4'. The reason is that none of the two sessions is immediately available as well as provides a just quality of service. If the session occurs in a club domain, the system will use the S3 that is able to meet the "Real-time" requirement of the session involvers. If the session occurs in a friend domain where the "Performance" requirement applies, the system will postpone the S4' until the expected receiver becomes available. If the session occurs between family members who have an "Easy" requirement on communications, the system then makes decisions according to the length of period it awaits for the S4' receiver to become available. If the waiting period is short, the system will use the S4'; if the period is long, it immediately uses the S3 (details referring to section 6.2.4).

A complicated scenario worth of mentioning here is that two users may have more than one type of relations at a time. For example, the husband and wife – Lisa and Bill – are colleagues as well. Then the ideal policy for this scenario is to guide the two users' communication activities according to their communication status (i.e., physical location and social relations). When determining the most appropriate connecting manner for a call from Lisa to Bill, the family relationship prevails at home (i.e., 5pm – 8am next day) whereas the colleague relation overtakes at work (i.e. 8am to 5pm). In the current version of software realization, we only consider a single relation between any two users in the communications network.

5.3.5 Procedure of Determining an Optimal Session Delivery

Summarizing the above four sections, the procedure of determining an optimal delivery manner for an expected session mainly takes four steps as shown in Figure 5-11.

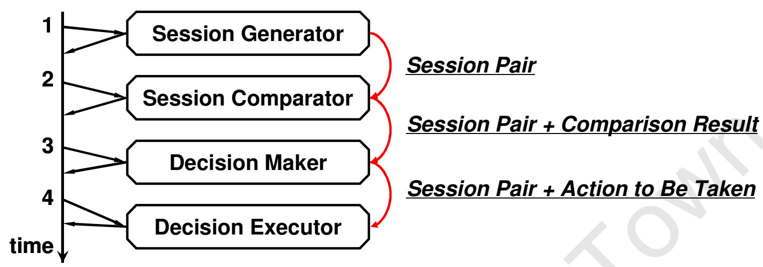


Figure 5-11. Major steps for determining an optimal session-delivery manner.

The session-delivery determination in Figure 5-11 includes three major steps. The first is to generate several pairs of virtual sessions for the expected one in Session Generator and send the generated session pairs to the next module (Figure 5-12). The second is to compare the virtual-session pairs in Session Comparator, with a set of comparison result achieved for each pair, and send these result sets to the next module (Figure 5-13). The last is to identify a session pair that provides the most reasonable solution for the expected session at the investigating moment and determine an optimal action to handle the identified session pair (Figure 5-14).

For a real communication session

Users	Vector of expected session and created sessions.				
Initiator and Receiver 1	Expected Session 1	Created Session 1.1	Created Session 1.2	...	Created Session 1.N_1
Initiator and Receiver 2	Expected Session 2	Created Session 2.1	Created Session 2.2	...	Created Session 2.N_2
...					
Initiator and Receiver M	Expected Session M	Created Session M.1	Created Session M.2	...	Created Session M.N_M

Figure 5-12. Generating virtual-session pairs in the Session-Generator module.

The generating scheme exhibited in Figure 5-12 aims to enumerate the entire potential sessions for an expected session by considering all potential receivers and their individual feasible sessions. According to the user-based session delivery (section 4.2.3), an expected session may be available for M potential receivers, referring to the first column in Figure 5-12. The "Receiver 1" is the originally wanted receiver in general. Then, according to the user-based session delivery (section 4.2.3), each receiver is able to propose $N_i, i \in [1, M]$ possible sessions for the expected one, referring to each row of data.

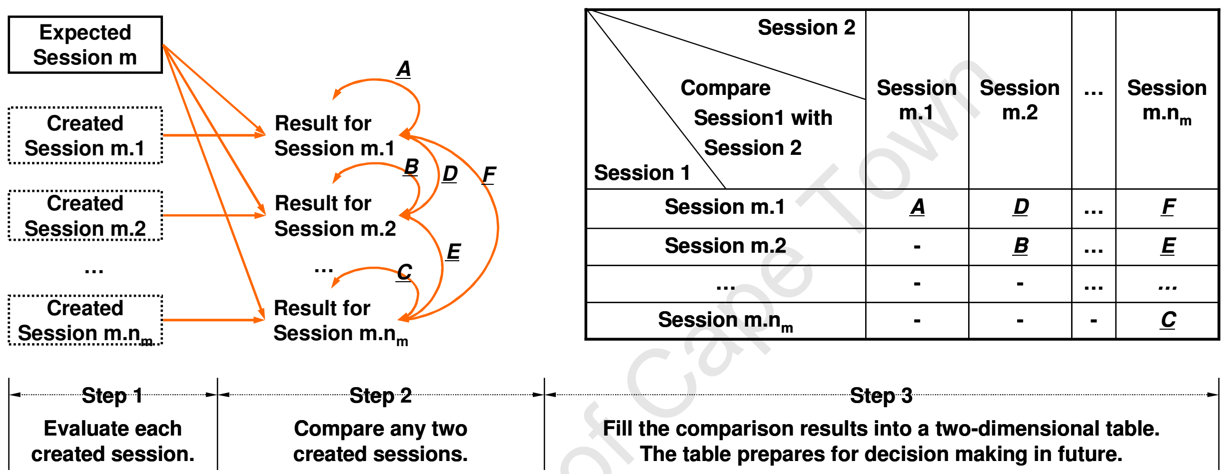


Figure 5-13. Comparing each pair of virtual sessions in the Session-Comparator module.

The comparison scheme depicted in Figure 5-13 compares the expected session with all the potential sessions proposed for one receiver. That is, it identifies the most suitable generated session in each row of Figure 5-12. Session Comparator first fetches a row of data from the table shown in Figure 5-12. The data contains the expected and all practical virtual sessions for a system-suggested receiver (step 1). The Comparator then compares each generated session with the expected one in terms of service performance, real-time availability of potential device, and the relationship of the potential receiver with the original initiator (step 2). Thereafter, the Comparator puts all sets of comparison results individually as row names and column names in a table and compares any two sets of results (step 3).

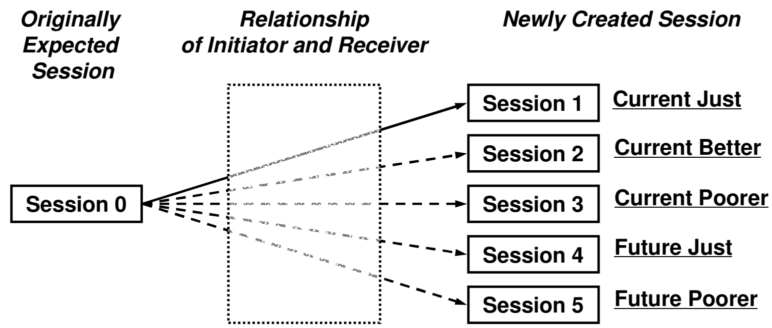


Figure 5-14. Identifying optimal session pair in the Decision-Maker module.

Finally, Decision Maker simply needs to identify the generated session that provides the best set of results in the comparison table. By considering the relationship strictness of the original initiator and the suggested receiver, it further determines the most feasible manner for the identified session. Figure 5-14 illustrates such a cooperative effect from the service performance and the social-relationship type.

5.3.6 Trust-related Issues Facilitating the Determination of Session Delivery

Using human nature to facilitate session processing is an effective way to apply human-like intelligence to the communications network. We select users' trustworthiness and their inter-trust degree as specific human natures to help determine an optimal session-delivery manner in the virtual-user system (sections 4.2.4.2 and 4.5.1).

We construct the following scenario to help explain the determination of session delivery using trust-related issues. User Ann called user Bill in Domain1 whereas Bill could not accept the call for some reason. The system then investigated on whether the third user Cora is able to receive the call concerning its social issues. Figure 5-15 exhibits the investigation procedure of determining whether the session is deliverable to Cora or not and, if yes, identifying the pre-conditions for a successful delivery.

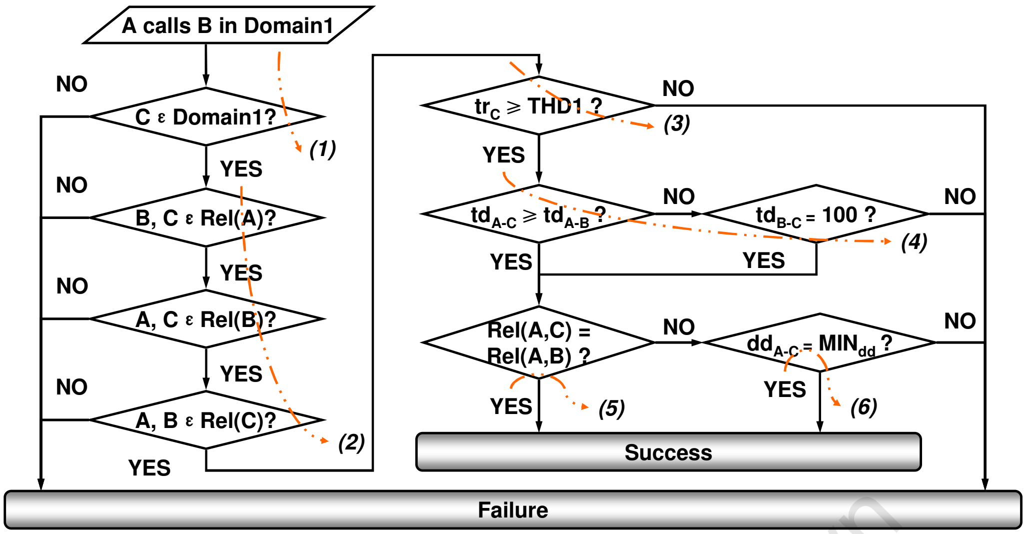


Figure 5-15. Identifying a qualified assistant user using trust-related issues.

"A", "B", and "C" individually stand for the users "Ann", "Bill", and "Cora". "C ε Domain1" means that Cora socially behaves in the Domain1. "B, C ε Rel(A)" means that both Bill and Cora are in Ann's relationship list. "tr_C" represents Cora's trustworthiness value. "$td_{A\text{-}C}$" represents the trust degree of Ann to Cora, i.e., the degree that Ann trusts Cora. "Rel(A, C)" stands for the relationship type of Ann and Cora. "$dd_{A\text{-}C}$" stands for the demanding degree from Ann to Cora. "MIN_{dd}" stands for the minimum closeness of any two demanding-degree values.

Figure 5-15 exemplifies the rules of identifying a trustable assistant user for an unsuccessful session, by judging whether Cora is worthy of trust in receiving the call from Ann for unavailable Bill. (1) The system first checks whether Cora socially relates to Ann and Bill in their common domain, Domain1. (2) If Cora is in the domain, the system then checks whether the three users are in each other's relationship list. If they are, they must have obtained a certain amount of mutual trust of each other from previous communication events. (3) Other than being trusted by the other two users, Cora should also meet the Domain1's minimum requirement on absolute trustworthiness, which generally mirrors a user's public reputation. (4) Then, unless the originally expected receiver – Bill – has complete trust on the newly suggested receiver – Cora – with a trust degree of "100", the extent to which Ann trusts Cora should be higher than that to which Ann trusts Bill. The reason is that the system has already sacrificed the originally expected receiver, so it should at least propose a user that the original initiator trusts enough to initiate a session. (5) If more than one user is able to meet the requirements, the system selects the one who has the same relationship with the initiator as that of the originally expected receiver with the initiator. (6) If no user has a matching relationship as that of the original receiver to the initiator, the system selects the session that the initiator demands most urgently to deliver to the suggested receiver. Lastly, if Cora fails to pass any of the above six checks, it is then not a qualified potential assistant receiver and is therefore unacceptable for the system.

The scenario constructed in Figure 5-16 gives an example for exhibiting the above rules.

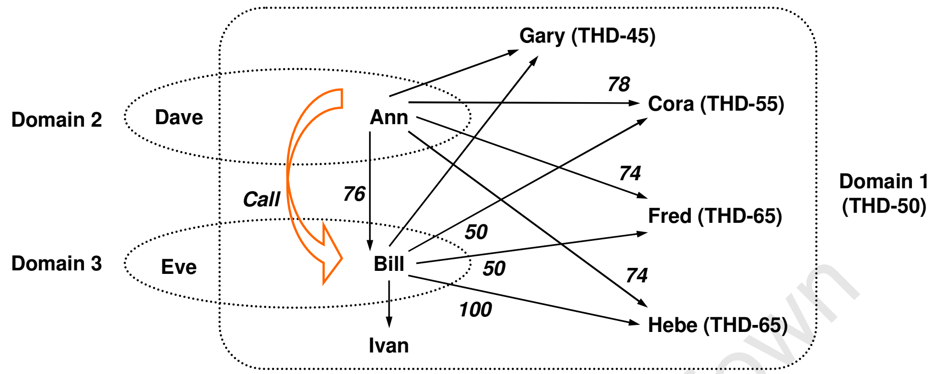


Figure 5-16. Example of identifying a trustable assistant receiver.

The sign "THD-xx" indicates the absolute trustworthiness value of the user next to the sign. Especially, the minimum trustworthiness requirement of Domain 1 is "50". The number besides the arrow indicates the trust degree from the user from whom the arrow starts to the user to whom the arrow points.

Figure 5-16 depicts the social topology of nine users, some of whom with trust degree and trustworthiness marked. The Domain1 requires minimum "50" on the trustworthiness value of a user if he/she participates in any session in the domain. Ann and Bill are socially related in the common domain – Domain1 – and they have their respective social domains Domain 2 and Domain 3. Ann started a call to Bill and Bill could not receive it. If the determined remedial manner is to identify a trustable user to assist Bill with the call, the system will then carry out the following steps concerning trust issues. Firstly, it identified that the five users of Cora, Fred, Gary, Hebe, and Ivan share the Domain 1 with Ann and Bill. Then the system excluded Ivan because she is not in the social relationship list of Ann. Gary was also identified unqualified because his trustworthiness value is lower than the minimum threshold of Domain 1. Among the rest, Ann trusted Cora more then she trusted Bill and thus Cora might be one of the qualified assistants. Fred and Hebe undertook a lower trust degree from Ann and they were not supposed to receive the call. However, Bill completely trusted Hebe in dealing with all his communication sessions (trust degree from Bill to Hebe is "100") and thus helped upgrade Hebe's trustworthiness in Ann's eyes. Hebe was an alternative qualified assistant. Whether Cora or Hebe became the final selected assistant receiver depends on the closeness of their relationships with Ann and the demanding degree of Ann to them on their sessions.

5.3.7 Self-learning Through User Social Relationships

When a user receives a session that it does not know the way to process, it then learns from its commonly related users with the initiator about the processing method. The system confines the searching scope for helpful related users within two degrees to reduce workloads. Figure 5-17 clearly exhibits a two-degree connection between the initiator and the receiver.

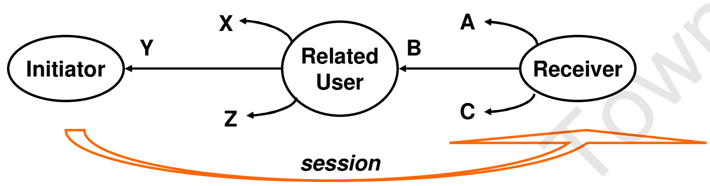


Figure 5-17. Learning communicating abilities through users' social relationships.

Two principles apply to the learn process illustrated in Figure 5-17. The first is the backward-search principle. When a session reaches the expected receiver, the receiver may fail to identify a proper delivery for the session on its own such as the receiver realizes that the initiator is not in its contact list. If so, the receiver then searches its related users to investigate whether one of its related users has a connection with the original initiator. The searching process is in an opposite direction of the session process. The second principle is the trust-incompletely-transfer principle. That is, if the receiver trusts some related user and the related user trusts the initiator, the initiator is then worthy of certain trust from the receiver through this trust-transfer procedure. Therefore, once the receiver can find a related user who relates to the initiator, the system assumes that the receiver trusts the initiator to a certain extent and correspondingly proposes a manner to set up a connection between the two users.

5.3.8 Quantifying Application Performance into Levels of Characteristics

Quantifying the continuous value range of an application characteristic into countable levels has three advantages. (1) It is easy to create a new application by choosing a set of values from the existent value levels and applying them to the respective characteristics. (2) It is easy for the Session-Comparator module to analyze the performance of an application by comparing the value levels of each characteristic. (3) If entering these characteristics and their respective values in a table, it is then easy to add new characteristics and values or delete unnecessary ones by simply inserting or deleting columns and rows. It is also easy to update the cell values of the table.

Table 5-11 gives an example of such a table by listing the characteristics accommodated by the HIFGN project. In the real communication world, the table is more complex and more comprehensive in both characteristic types and value levels as can be expected.

Table 5-11. Value levels of application characteristics in the HIFGN project.

Level	Service Type	Data Type	Quality			Security	Billing		
			Bandwidth	Packet Delay	Bit Error Rate	Availability	Type	Cost	Charged Party
1	Email	Data	0~8 KHz	0 ms	0%~0.00001%	Available	ChargeToUnit	0	Initiator
2	FileSharing	Audio	8~56 KHz	0~1 ms	0.00001%~0.0001%	NotAvailable	ChargeToUser[Δ]	1	Receiver
3	VoiceCall[Δ]	Video	56~64 KHz	1~10 ms	0.0001%~0.001%	ToBeAvailable[Δ]	ChargeToSession	3	Both
4	VoiceMessage	DataAudio[Δ]	64~128 KHz	10~100 ms	0.001%~0.01%		ChargeToZero	5	None
5	Messenger	DataVideo	128~256 KHz	100~1000 ms	0.01%~0.1%			10	
6	Video	AudioVideo	256~512 KHz	1~10 s	0.1%~1%				
7	Teleconferencing	DataAudioVideo	512~1024 KHz	10~100 s	1%~10%				
8			1024~2048 KHz	100~1000 s	10%~100%				
9			2048~4096 KHz	1000~ 1000000000 s					
10			4096~8192 KHz						
11			8192~10000 KHz						
12			1 MHz						
13			10 MHz						
14			100 MHz						
15			1 GHz						
16			10 GHz						

[Δ] E.g., "VoiceCall" implies that the service type of the application is a voice call.

[Δ] E.g., "DataAudio" implies that the application carries both data and audio data.

[Δ] E.g., "ToBeAvailable" implies that the devices for the application will become available in the near future but still within the set-up time for the application.

[Δ] E.g., "ChargeToUser" implies that the cost on running the application will be charged to the session involvers.

5.3.9 Searching Necessary Information from Databases

The procedure of obtaining application information from the application database and that of obtaining user information from the user database are the same, as shown in Figure 5-18:

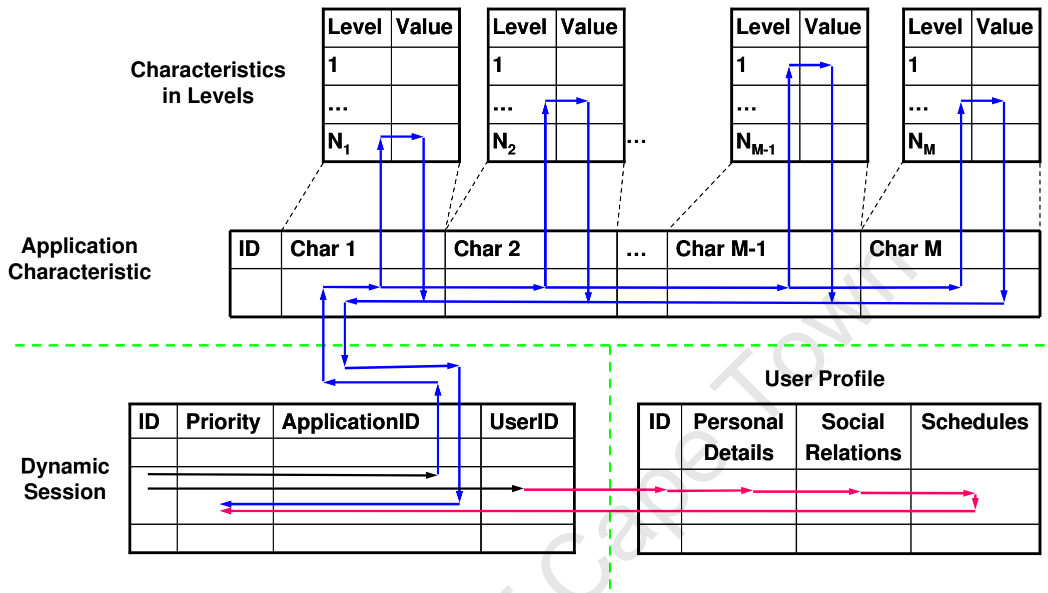


Figure 5-18. Searching for information from relevant databases.

Figure 5-18 illustrates the way that the system obtains different types of information from the respective databases. The virtual-user system brands each type of information with a globally unique ID and the databases store the specifications of information using the ID as index. To successfully locate the specific data area of a relevant database for the request for a type of information, the system composes the session element with various IDs respectively for all types of information. For example, it identifies the characteristic types for an application through the application ID and further identifies the values for each type through the level ID. The system also identifies a user's personal details, social relationships, and weekly schedules through the user ID.

5.4 Tentative System Test

The HIFGN project is a software realization of the virtual-user system that implements the hypothesis of introducing human-like intelligence into the communications network.

Therefore, the comprehensive and effective testing of the project will prove and validate the hypothesis and its corresponding virtual-user proposal. There are three ways to testify to the project.

(1) Evaluate the functionality of the system modules. These modules are made up in programming languages, aimed at applying intelligence to the network. Therefore, their proper and efficient functioning can prove the system effective as well as applicable to the real world.

Methods of testifying to the functionality include popping out visible information windows in Netbeans to indicate the action carried out for session delivery, running different communication scenarios, and collecting corresponding session-delivery results (Chapter 6). By comparing them using common knowledge, we can draw conclusions on the advantages and disadvantages of operating these modules and further determining whether the modules are able to implement the hypothesis.

(2) Evaluate the selected network-recognizable human characteristics (section 3.1.4). Before evaluation, the system first needs to quantify abstract human characteristics, with the resulting dataset supposedly capable of imitating and reflecting any change in the characteristics. The key steps to quantify a human characteristic include identifying the value range and changing trend of the characteristic, sampling at rates proportional to the density of changes in value, and assigning each sample an appropriate value from the range. Quantifying human nature is complex and difficult due to the imprecise range of usable values, the shortage of references, the inconsistency in trends, and the impracticability of evaluating the sample dataset.

The human characteristic – trustworthiness – is chosen to represent user sociability in the project. By quantifying the trustworthiness as a natural number between zero and a hundred and applying this absolute trustworthiness value to each network user, the changes in users' trustworthiness are expected to affect the manner and success rate of communication-session delivery.

(3) Evaluate system performance. In the final analysis, whether the virtual-user system is considered of any value depends on whether network users would love to use it. Users will only use a system that can bring them great benefits and serve them as their shadows. In this

regard, the better performance the system can provide, the more competitive it is with other communication systems.

The performance parameters needed for evaluation in the project include session-delivery success rate, fluctuation degree of network traffic, communication-resource utilization, effect of human characteristics on communication sessions, and permitted time delay during session set-up (Chapter 7).

As suggested earlier (section 5.2.4), we import the function module Session Simulator in the HIFGN project to generate communication traffic for system test.

5.4.1 Session Simulator

The function module Session Simulator generates a series of communication events at a rate to simulate real communication traffic. It is regulated in the package "hifgn.module.sim" as shown in Figure 5-19.

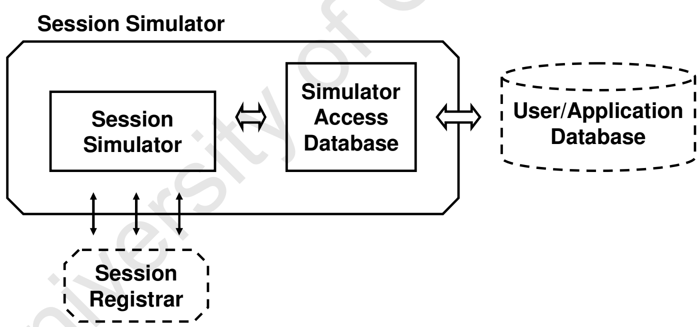


Figure 5-19. Layout of the classes in the "hifgn.module.sim" package.

The Session-Simulator class is responsible for simulating real-world communication traffic that serves as the input for the virtual-user system. Session Simulator first generates traffic by considering distribution type of event number over time, complete traffic duration, traffic start time, data type, event-occurring rate, and execution period of each event. For testing purpose, the module fetches necessary information from the information databases to generate a traffic that is surely recognizable by the virtual-user system. After having generated the traffic, the module then invokes the Session-Registrar module to start processing the traffic. The Session-Simulator classes only communicate with Session Registrar.

Although there are many on-going studies on social relations with obtained real-world communication traces, the tracing data are not directly usable in the virtual-user system because of the incompatible data format. Picking out useful data and quantifying them as system-recognizable data are out of the scope of the work.

5.5 Summary of Chapter 5

This chapter narrated the HIFGN project that realizes the virtual-user system in a software environment. It first proposed using Java programming language to implement system components and MySQL database tool to store user- and application-information. It then introduced the software design of system-parameters, elements, and modules in terms of needed variables and layout of used Java classes. Thereafter, the essential mechanisms that embodies human-like intelligence were described, with each performed by the cooperation of system elements and modules. Finally, the chapter addressed three methodologies of testifying to the project in theory. Noticeably, since our focus is on the proof-of-concept of intelligence establishment in the network, more advanced and comprehensive studies such as a scalability analysis of the system for a large telecommunication network or a quantitative measurement of the effect that intelligence exerts to communications will be deferred to future study.

Chapter 6 will testify to the effectiveness of the programmed virtual-user system by showing whether the system works functionally, how it realizes human-like intelligence, and how it absorbs social knowledge. Corresponding explanations are given to all testing results.

Chapter 6 Validating the Virtual-user System

After the structural design and software implementation, the next important step is to verify the functionality of the virtual-user system. Positive test results will prove the realizability of the system and further prepare it for the prototype of academic product.

The selected test methods include validating the major function modules of the virtual-user system (section 6.1), testing whether the system is able to determine the most appropriate session-delivery manners for different communication scenarios (section 6.2), and demonstrating how the specifically selected intelligence characteristic – social relationship – assists in making decisions (section 6.3). A few other user-friendly features make the system easily accessible to network users and service designers (section 6.4). Despite those advantages, the software implementation of the system still has several limitations (section 6.5).

We describe the simple topology of Yang's social relationships in Figure 6-1. The validation and the corresponding analysis of the system below are all based on this topology.

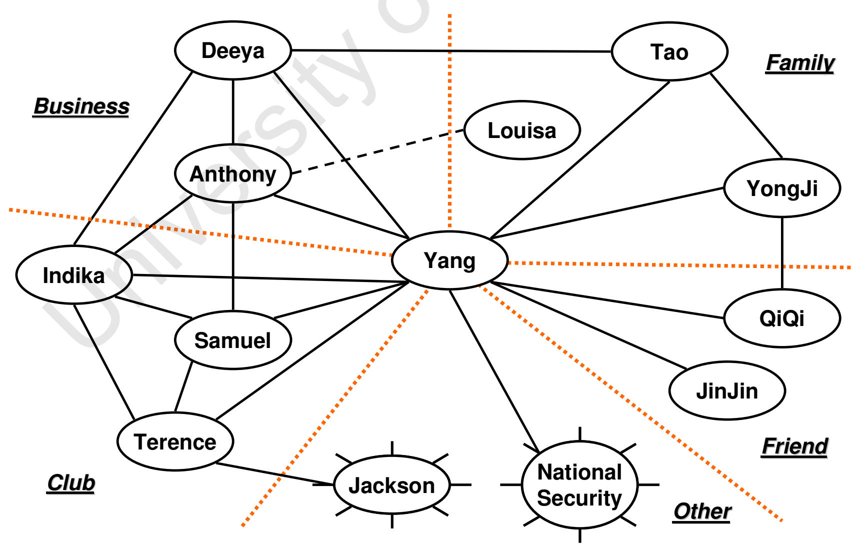


Figure 6-1. An illustration of Yang's social-relationship topology.

In Figure 6-1, Yang's social life spans five social domains: "*Family*", "*Business*", "*Club*", "*Friend*", and "*Other*". Each user relates to Yang in a unique type of relationship that locates in a specific domain (Table C-3). Some users are directly connected.

6.1 Effectiveness of the Function modules

In theory (Chapter 4), the virtual-user system is able to determine the most successful session-delivery manner, regardless of the session involvers' communication status and the network's traffic status. To determine such an optimal manner, the system first registers a real communication event as a real session that is recognizable, readable, and modifiable by the system (section 6.1.1). Then it generates several pairs of virtual sessions for the real session (section 6.1.2) and compares the two virtual sessions of each pair (section 6.1.3). Afterwards, the system selects the pair with the best service performance (i.e., a combination of real-time availability and quality of service) and determines an optimal action to carry out the practical session in the selected pair (section 6.1.4). Finally, the system instructs the relevant modules to perform the determined action (section 6.1.5). When needed, the system stores the temporarily failed sessions until they become deliverable (section 6.1.6).

Before validating the virtual-user system, we first assume that the system receive communication events carried in messages with a fixed data format from the existing network, such as receiving a Session-Description-Protocol (SDP[47] [100]) message from an IP Multimedia Subsystem. The system then interprets the event information as the system-acknowledged information, including the expected application ID, the initiator ID, the originally expected receiver ID, and the set-up time that the system started processing the event. The system assigns a unique real-session ID to each event and uses the real-session ID to connect all the virtual sessions relating to the event.

We further construct a communication scenario in Table 6-1 and, by operating a communication case described by the scenario, validate the functionality of system modules.

Table 6-1. Communication case for illustrating a successful session delivery.

Initiator	Receiver	Application	Session Start Time	Expected Manner
Yang	Anthony	VoiceCall_Phone	2007-08-10.14:00:01	Deliver

[47] SDP is a format for describing streaming media initialization parameters.

Table 6-1 provides the information to construct a communication scenario where Yang called Anthony at his office phone at 14:00:01 on 10 August 2007. According to the users' communication statuses and preferred session-processing rules, the system should be able to determine to "*deliver*" the call.

In addition, we clarify three major stages of handling on-going sessions in the system, including the session-processing, session-reprocessing, and session-executing stages (Figure 6-2).

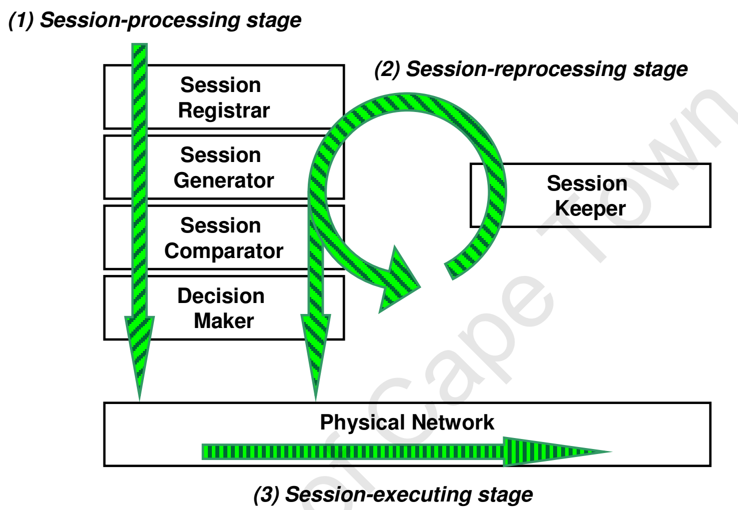


Figure 6-2. Stages of session processing, session reprocessing, and session executing.

Figure 6-2 illustrates the three possible stages of sessions during their stay in the system. In the session-processing stage (the arrow marked by right-upward diagonals), the system has just registered the session and performs the session-generating, comparing, and decision-making for the first time. In the session-reprocessing stage (the arrow marked by left-upward diagonals), the system has determined to postpone the session and repeats the process of generating, comparing, and decision-making only when necessary. In the session-executing stage (the arrow marked by vertical lines), the system has successfully determined the final session-delivery manner and instructs the physical network to execute the session in reality.

6.1.1 Step 1: Registering a New Real Session in Session-Registrar Module

Registering a new session in the virtual-user system occurs in Session Registrar. The sequential steps include creating a new session element, informing Application Library to fetch the application information from the application database, informing Virtual Personal Profile to fetch the information of the initiator and the receiver from the user database, and updating the initiator's most recent communication status. On completing these actions, the system should have obtained the basic information of the session.

The Registrar then generates a session element that contains the index information of the real session, comprising several types of IDs. Figure 6-3 shows a simple version of Java run-time output regarding such information.

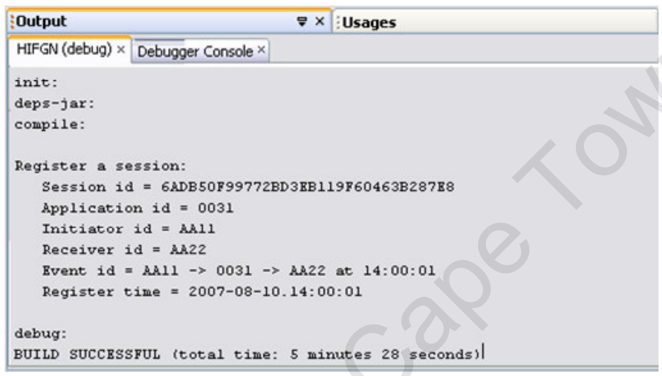


Figure 6-3. Java output on the completion of registering a session.

The build-successful-total-time of "5 minutes 28 seconds" includes the time of system operating and information collecting, such as printing the screen and pasting the figures, so is with the following figures.

Figure 6-3 displays the session information that the system obtains from a real communication event. The event shows Yang (user ID "AA11") as the initiator, Anthony (user ID "AA22") as the receiver, and VoiceCall (application ID "0031") as the application. The uniquely acknowledged real-session ID is "*6ADB50F99772BD3EB119F60463B287E8*". The system started processing the event at 14:00:01 on 10 August 2007. In theory, the event ID is supposed to be the identification of the event in the physical network such as the session identifier[48] in SDP. Nevertheless, the ID value does not affect the decision-making results about the event-processing manner in essence. We therefore simply use a descriptive sentence about the

[48] The session identifier in Session Description Protocol is a combination of user name, session id, network type, address type, and address. It is globally unique.

event to represent the event ID. The information shown in Figure 6-3 together means that Yang initiated a voice call to Anthony at 14:00:01 on 10 August 2007.

After having generated the session element using the interpreted event information, the system stores the session as a piece of record in the relevant database (Figure 6-4) and keeps track of the session information implied by the record from that moment on.

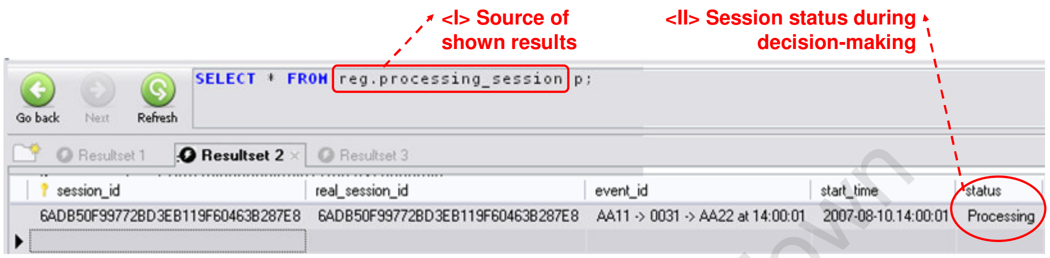


Figure 6-4. MySQL results on the completion of registering a session.

Figure 6-4 displays a record about the newly registered session in the MySQL database. The record resides in the "*processing_session*" table of the "*reg*" scheme in MySQL database (mark <I>). The system uses the first session ID generated during registration as the uniquely acknowledgeable "*real session id*" for the event in all system modules. That is, if any module needs to generate new virtual sessions for the event in the future, it always refers to this real-session ID as the unique index of the event. The "*status*" describes the real-time process status of the session (mark <II>). It has two optional values: the "*processing*" indicates that the session is in the session-processing stage and the "*suspending*" in the session-reprocessing stage.

6.1.2 Step 2: Generating a Virtual-Session Pair for the Real Session in Session-Generator Module

After session registration, Session Generator generates several pairs of virtual sessions for the registered session. Each pair is composed of two session elements, one for the expected virtual session and one for the system-identified practical session. The Generator mainly carries out the following three steps to complete a session generation.

(1) The Generator first composes several pairs of session elements and copies the key information of the registered session into the corresponding data fields of these elements.

(2) It then obtains relevant application- and user-information from the databases and fills in the element for the expected session in each pair with the obtained information. Figure 6-5 illustrates the basic information of a session element, whether the element is for an expected or newly created session.

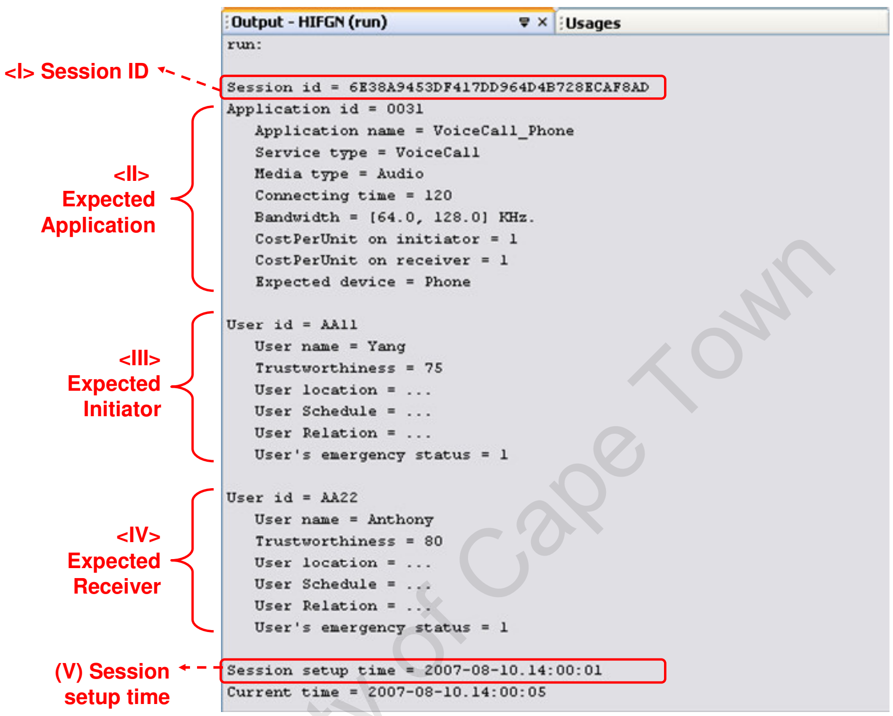


Figure 6-5. Basic session information of a filled-in session element.

Figure 6-5 exhibits the key information of a session element that contains the expected virtual session under the communication scenario described in Table 6-1. The randomly generated "*session id*" uniquely identifies the virtual session in Session Generator (mark <I>). The element has also recorded the "*session set-up time*" (mark <V>).

The area of expected application shows that the system has obtained abundant information from the application database through the known application ID "0031" (mark <II>). The "*service type*" identifies the service that the application can provide, the "*media type*" indicates the communication media that the application uses, the "*connecting time*" determines the lifetime allowed for application set-up (i.e., session set-up) in the system, and the "*bandwidth*" restricts the quality of service. Furthermore, the "*cost per unit*" indicates the charging standard to the users and the "*expected device*" specifies the device that is most suitable for performing the application.

The area of expected initiator (mark <III> in Figure 6-5) and that of expected receiver (mark <IV> in Figure 6-5) respectively describe the initiating user and the receiving user in the same format. Figure 6-6 uses Yang's communication profile as an example to illustrate the detailed user information.

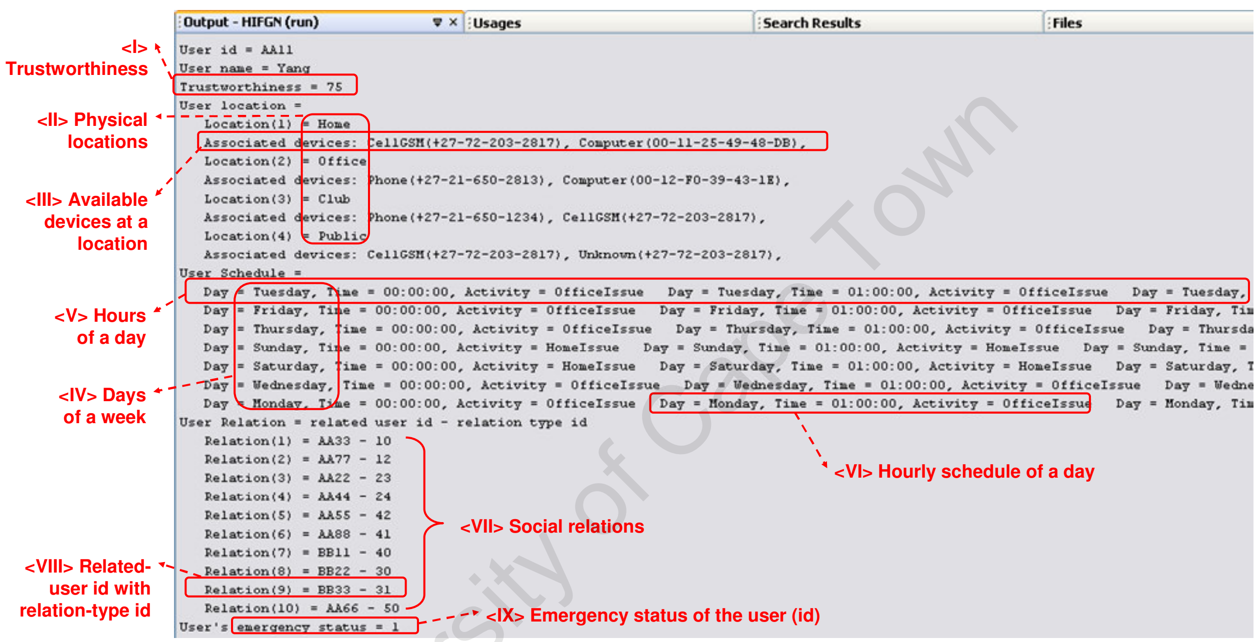


Figure 6-6. Detailed user information in a filled-in session element.

Figure 6-6 provides a full description of Yang's personal communication profile, uniquely identified by Yang's user ID "AA11". The "*trustworthiness*" value "75" indicates how much Yang is worthy of trust in all communication sessions in which she participates (mark <I>). Yang's "*user location*" contains two types of information: one is the physical locations where she usually frequents (mark <II>) and the other is the devices available to her at each location (mark <III>). Yang's "*user schedule*" is a two-dimensional timetable with the day of a week as a row name (mark <IV>), the hour of a day as column name (mark <V>), and her social activity in each hour of a certain day as cell value. For example, the piece of schedule information (mark <VI>) indicates that Yang is busy with her office issue for one hour starting from 01:00:00 on Monday. Yang also maintains an admitted contact list of 10 users with each having a specific relationship to Yang (mark <VII>). For example, the ninth relationship to Yang shows that Yang relates to a user with user ID “BB33” in a relationship with type ID “31” (mark <VIII>), implying that JinJin is Yang’s common friend (Table C-3). When a user’s emergency status is emergency, his/her “*emergency status*” is marked as “0”, otherwise “1”. Therefore, all calls that Yang starts or receives are normal calls with no need for emergency process (mark <IX>).

(3) After having filled in the detailed information for the expected virtual session, the Session Generator identifies all potential applications that are technically available for the expected session (section 5.3.2). A proposed technically available virtual session is able to provide the same type of service as that of the expected one using a different connecting manner (i.e., different devices). The Generator then sets up a session element for every newly identified virtual session. After filling the session elements with the information of potential applications in the same way as it does to the expected one, it will associate each newly created session with the expected one as a pair. Figure 6-7 illustrates the scenario where the system has identified five applications that are technically available for the voice call in Table 6-1.

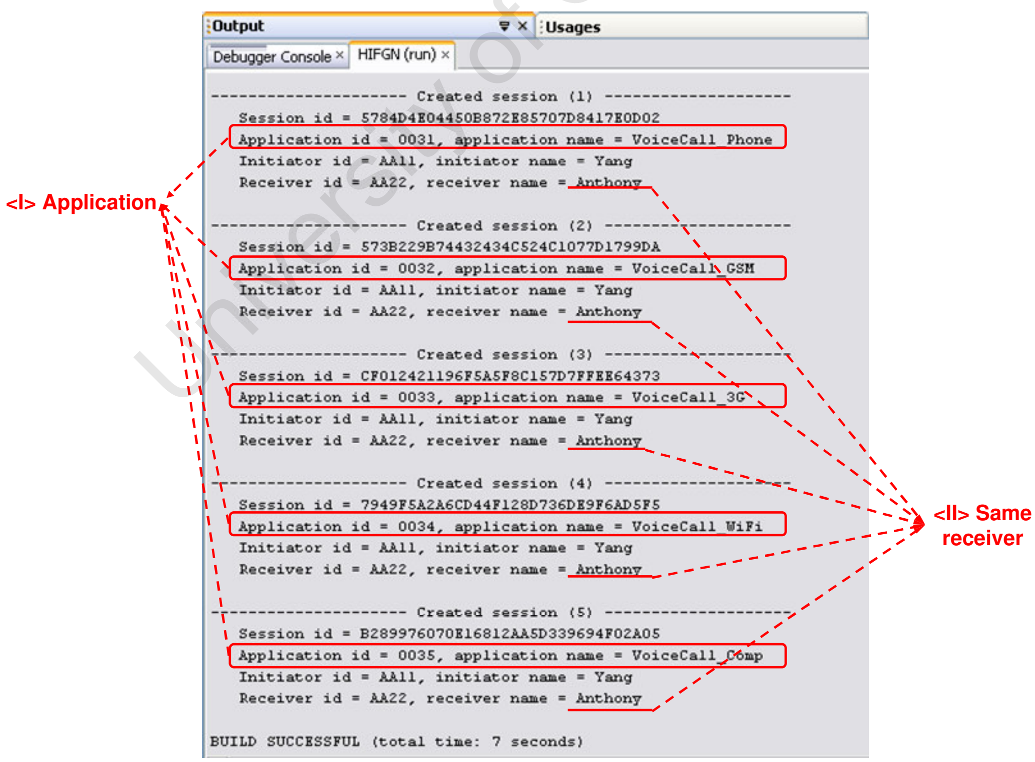
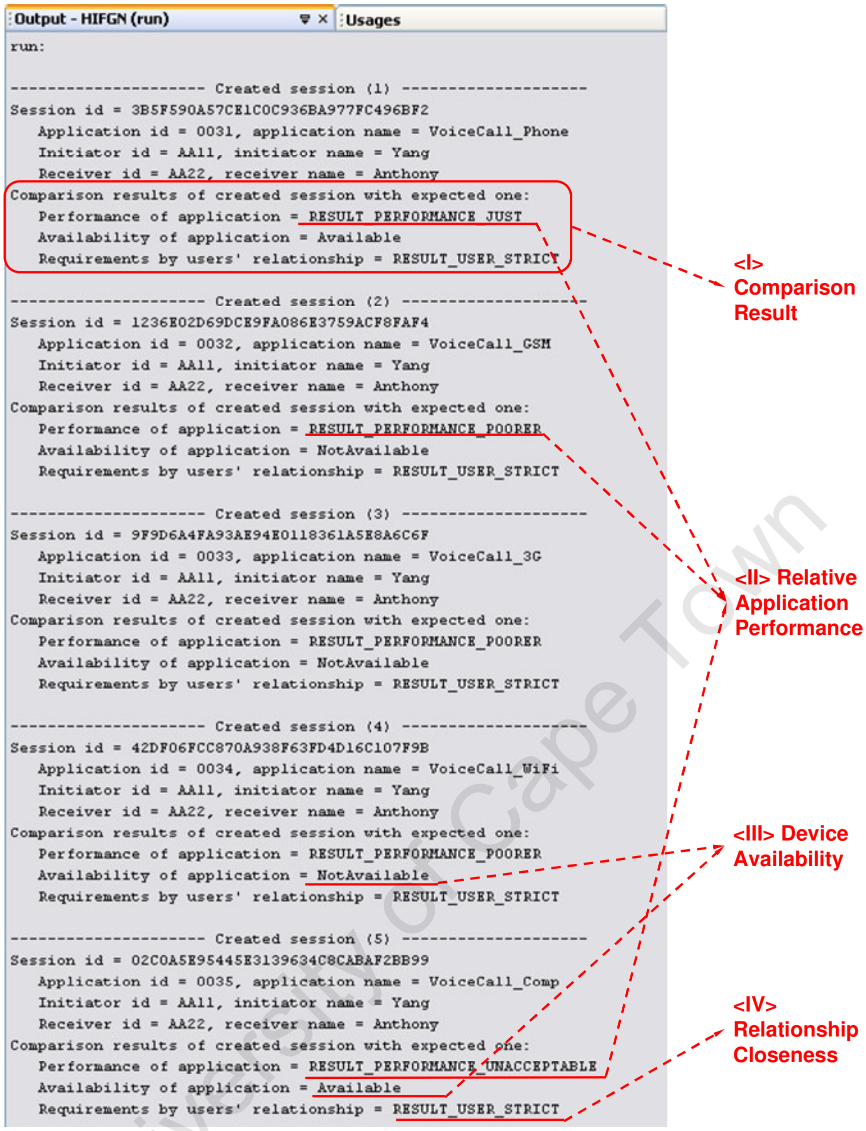


Figure 6-7. Simplified information of all identified potential sessions for the expected one.

The five applications in Figure 6-7 are all able to provide the voice-call service yet on different devices, including "*phone*", "*GSM*" cell, "*3G*" cell, "*WiFi*" cell, and "*computer*" (mark <I>). The potential receivers can be different in theory, whereas it turns out to be the same user Anthony in this case (mark <II>).

6.1.3 Step 3: Comparing Virtual-session Pair in Session-Comparator Module

On receiving several virtual-session pairs from Session Generator, Session Comparator compares the created virtual session with the expected one in each pair. Figure 6-8 displays the comparison results of the five potential applications for Yang's voice call to Anthony.



```
Output - HIFGN (run)    Usages
run:

-------------------- Created session (1) -------------------
Session id = 3B5F590A57CE1C0C936BA977FC496BF2
   Application id = 0031, application name = VoiceCall_Phone
   Initiator id = AA11, initiator name = Yang
   Receiver id = AA22, receiver name = Anthony
Comparison results of created session with expected one:
   Performance of application = RESULT_PERFORMANCE_JUST
   Availability of application = Available
   Requirements by users' relationship = RESULT_USER_STRICT

-------------------- Created session (2) -------------------
Session id = 1236E02D69DCE9FA086E3759ACF8FAF4
   Application id = 0032, application name = VoiceCall_GSM
   Initiator id = AA11, initiator name = Yang
   Receiver id = AA22, receiver name = Anthony
Comparison results of created session with expected one:
   Performance of application = RESULT_PERFORMANCE_POORER
   Availability of application = NotAvailable
   Requirements by users' relationship = RESULT_USER_STRICT

-------------------- Created session (3) -------------------
Session id = 9F9D6A4FA93AE94E0118361A5E8A6C6F
   Application id = 0033, application name = VoiceCall_3G
   Initiator id = AA11, initiator name = Yang
   Receiver id = AA22, receiver name = Anthony
Comparison results of created session with expected one:
   Performance of application = RESULT_PERFORMANCE_POORER
   Availability of application = NotAvailable
   Requirements by users' relationship = RESULT_USER_STRICT

-------------------- Created session (4) -------------------
Session id = 42DF06FCC870A938F63FD4D16C107F9B
   Application id = 0034, application name = VoiceCall_WiFi
   Initiator id = AA11, initiator name = Yang
   Receiver id = AA22, receiver name = Anthony
Comparison results of created session with expected one:
   Performance of application = RESULT_PERFORMANCE_POORER
   Availability of application = NotAvailable
   Requirements by users' relationship = RESULT_USER_STRICT

-------------------- Created session (5) -------------------
Session id = 02C0A5E95445E3139634C8CABAF2BB99
   Application id = 0035, application name = VoiceCall_Comp
   Initiator id = AA11, initiator name = Yang
   Receiver id = AA22, receiver name = Anthony
Comparison results of created session with expected one:
   Performance of application = RESULT_PERFORMANCE_UNACCEPTABLE
   Availability of application = Available
   Requirements by users' relationship = RESULT_USER_STRICT
```


Figure 6-8. Comparison results of each potential session with the expected one.

Figure 6-8 displays the comparison results of the five created virtual sessions with their expected session. The set of comparison results for a session pair includes the relative application performance of the created session to the expected one, the availability of the devices proposed in the created session, and the social-relationship closeness of the suggested session involvers in the created one (mark <I>).

The "*relative application performance*" signifies how well the newly created session meets the requirements by the expected one in terms of service performance. The types of fulfilment degree include "*just*", "*better*", "*poorer*", and "*unacceptable*", which individually mean that the created session is able to provide the same-, better-, poorer-, and unacceptable-quality of service when compared with the expected one. For example, the application with ID "0031" is able to provide the voice-call service for Yang, the one with ID "0032" can do so as well but with a poorer call quality, and the one with ID "0035" cannot provide the call service at all (mark <II>).

The "*device availability*" shows whether the proposed device in the newly created session is available for the suggested receiver at the investigating moment. It can be "*available*", "*to be available*", or "*not available*", which individually mean that the device is immediately available, will be available in the session connecting lifetime, and will not be available in the connecting lifetime for the receiver. For example, between the two observed created sessions, only the devices in the created session (5) are real-time available for their receiver (mark <III>).

The "*relationship closeness*" in the created session is the strictness degree of the requirements by the initiator on his/her sessions to the receiver. It can be "*strict*", "*performance*", "*real-time*", or "*easy*", which respectively mean that the initiator requires an immediately-available and just-quality service, only just-quality service, only immediately-available service, or no-special-requirement service with the receiver. It can also be "*no relation*", meaning that the initiator and the receiver do not have each other in their contact list. For example, the strictness degree of Yang and Anthony is "*strict*" because they communicate with each other in the business domain where neither downgraded nor delayed service is allowed (mark <IV>).

6.1.4 Step 4: Determining an Optimal Session-Delivery Manner in Decision-Maker Module

Based on the comparison results of all potential sessions with the expected one, Decision Maker identifies the potential session that provides the best service performance among all and further determines the way of processing it. Figure 6-9 illustrates the final selected potential session with the determined action for Yang's voice call to Anthony.

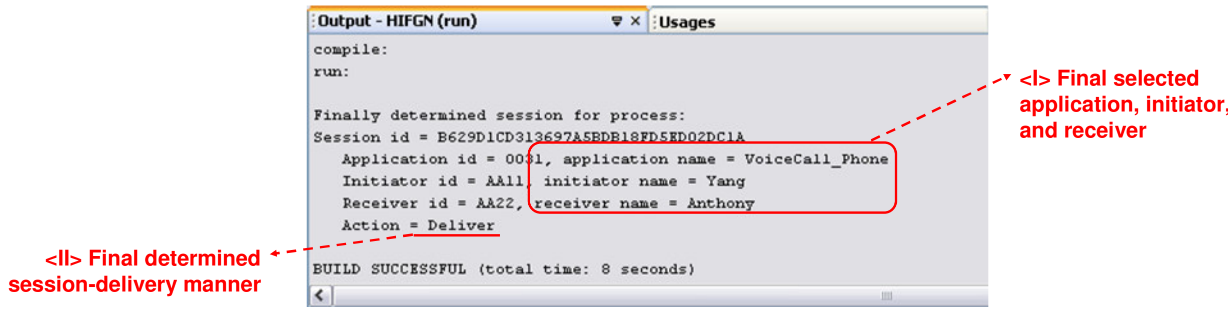


Figure 6-9. Decision-making results on the optimal session-delivery manner.

Figure 6-9 shows the decision-making results for Yang's voice call to Anthony based on the comparison results in Figure 6-8. The system suggests that Yang use the application with ID "0031" to immediately start the voice call to Anthony (mark <I>), because the session under such conditions possesses the highest probability to be successfully delivered in the physical network according to system diagnosis (mark <II>).

6.1.5 Step 5: Executing the Determined Session-Delivery Manner in Decision-Maker Module

After the system has determined an optimal created session for the expected one using the users' real-time communication status and social relationship, it instructs the relevant function modules to carry out the determined session. We describe each execution result in a sentence and surround it by stars in the output window. Figure 6-10 shows an example of a successful delivery.

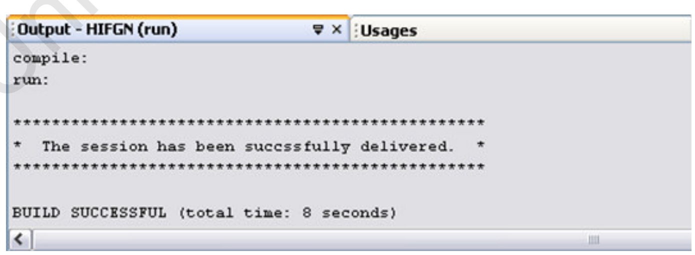


Figure 6-10. Java output on the completion of decision execution.

In addition, the system also records the final identified virtual session and the determined action in the database as shown in Figure 6-11.

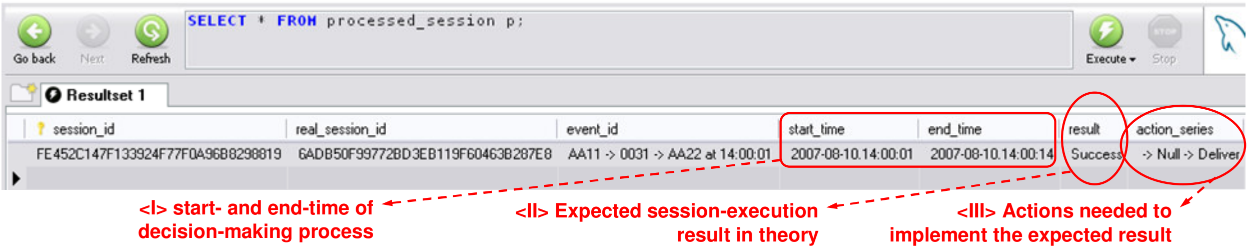


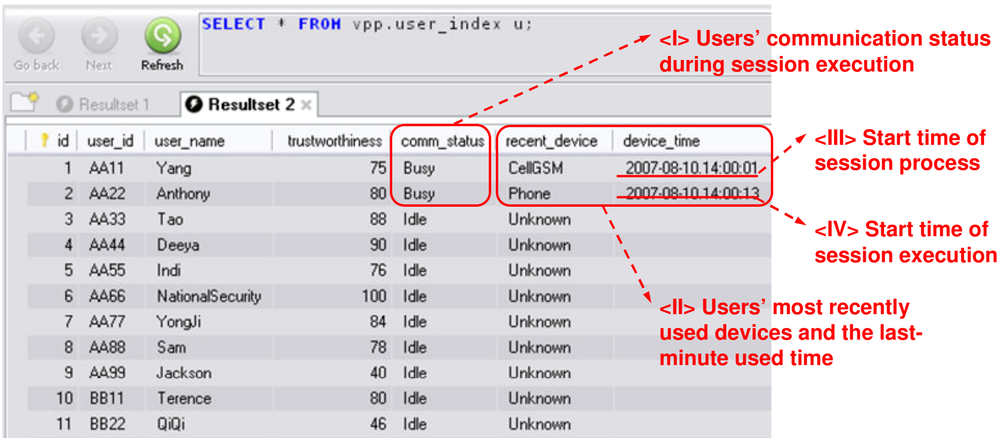
Figure 6-11. MySQL results on the completion of decision execution.

Some information in Figure 6-11 is helpful in analysing the session decision-making. The difference between the start-time and the end-time discloses that it takes 13 seconds for the system to determine the optimal solution for the expected voice call from Yang to Anthony (mark <I>). Note that a general session takes around one second to set up (Figure 7-15), much shorter than 13 seconds. The majority time of the 13 seconds are spent in capturing the figure. The "*result*" indicates that the system is eventually able to successfully deliver the expected session using some final determined delivery manner whatever actions are taken (mark <II>). The session-execution result can be "*Success*" or "*Failure*". The "*action series*" sequentially lists all the actions that the system has attempted for a successful session delivery (mark <III>). These actions can be *deliver*, *force*, *postpone*, *help*, *learn*, or *fail*. The first two actions will surely result in "*Success*" and the last one "*Failure*". The others can result in either "*Success*" or "*Failure*", depending on user communication profile and network traffic status.

In real-world communications, instead of displaying the determined execution results in Java- and MySQL-output windows as above, the system takes the initiative to instruct the physical network to execute the session based on the decision.

6.1.5.1 Example of Session Executing after Decision-Making

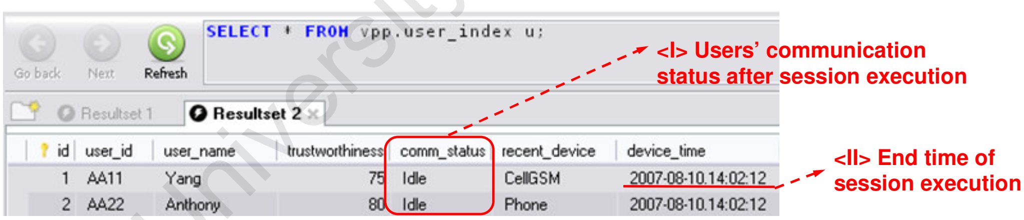
We real-time record session involvers' changing communication statuses in the database for two reasons. Firstly, being aware of user status helps determine the most possible delivery manner of the sessions with which the users are involved. Secondly, these recorded user statuses together with the preset user schedules provide an enhanced way of tracking users' real-time communication activities. Figure 6-12 and Figure 6-13 individually describe the scenarios where Yang is in a call with Anthony and where Yang has just finished her call with Anthony.



id	user_id	user_name	trustworthiness	comm_status	recent_device	device_time
1	AA11	Yang	75	Busy	CellGSM	2007-08-10 14:00:01
2	AA22	Anthony	80	Busy	Phone	2007-08-10 14:00:13
3	AA33	Tao	88	Idle	Unknown	
4	AA44	Deeya	90	Idle	Unknown	
5	AA55	Indi	76	Idle	Unknown	
6	AA66	NationalSecurity	100	Idle	Unknown	
7	AA77	YongJi	84	Idle	Unknown	
8	AA88	Sam	78	Idle	Unknown	
9	AA99	Jackson	40	Idle	Unknown	
10	BB11	Terence	80	Idle	Unknown	
11	BB22	QiQi	46	Idle	Unknown	

Figure 6-12. Users' real-time communication status (start of session execution).

Figure 6-12 illustrates the user communication stats when the system starts executing a session in the physical network. After the physical network starts executing the session, both the session initiator and the receiver will be busy with the session, such as Yang and Anthony being busy with the voice call (mark <I>). Yang's most recently used device was her GSM cell phone that she used to start the call at 14:00:01 and Anthony's most recently used device was his phone that he used to receive Yang's call at 14:00:13 (mark <II>). Although Yang had been busy since 14:00:01 (mark <III>), her real conversation with Anthony only started at 14:00:13 (mark <IV>).



id	user_id	user_name	trustworthiness	comm_status	recent_device	device_time
1	AA11	Yang	75	Idle	CellGSM	2007-08-10 14:02:12
2	AA22	Anthony	80	Idle	Phone	2007-08-10 14:02:12

Figure 6-13. Users' real-time communication status (end of session execution).

Figure 6-13 illustrates the user communication stats when the system starts executing a session in the physical network. We assume that each session execute for a fixed period. For example, a call executes for two minutes. Then, two minutes after the call started at 14:00:13 (mark <IV> in Figure 6-12), Yang and Anthony's real-time statuses changed back to idle (mark <I> in Figure 6-13). Meanwhile, the system updates the time when the two users last used their devices (mark <II> in Figure 6-13).

6.1.6 Step 6: Storing Suspended Session Pairs in Session-Keeper Module

If a session needs to perform the manners of *postpone*, *help*, or *learn* to achieve the "*success*" result, the system then suspends the session in the database for a certain period. Session Keeper is in charge of storing these suspended sessions and calling on other relevant modules to reprocess them whenever necessary.

We construct a communication scenario in Figure 6-14 to demonstrate the situation where a *postpone* action is necessary. We assume a normal session process for decision-making lasts for two seconds and each call executes for 12 seconds. It is noted that both the values are in accordance with reality and they together provide better illustration. Yang initiated a call to Anthony at 14:00:01 and the call completed decision-making at 14:00:03. From that moment on, Yang was in a call conversation with Anthony for 12 seconds. Later on, Louisa called Anthony at 14:00:11 when Anthony was still busy on the phone. The system then decided to postpone Louisa's call until Anthony changed to be available.

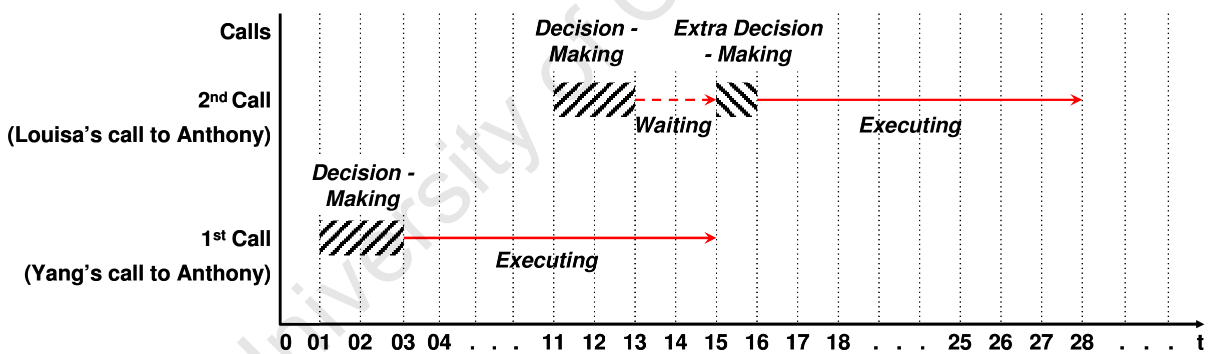
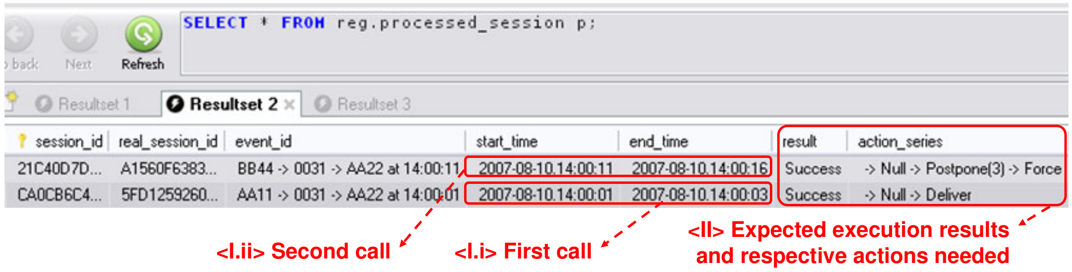


Figure 6-14. Session stages for two conflicting calls.

Figure 6-14 exhibits a theoretical procedure of the session-processing, reprocessing, and executing stages for two conflicting calls. The first call from Yang to Anthony initiated and proceeded normally, preparing Anthony's status to be busy. The second call from Louisa to Anthony started during the execution of the first one. After the general decision-making for the second call, the system identified that the expected receiver for the second call – Anthony – was still busy with the first call. It then suggested that the second call be suspended for a certain period. As soon as Anthony became available, the system re-evaluated the execution possibility of the second call and eventually instructed the execution of the second call after having confirmed its deliverability. The Java output of such a procedure is shown in Figure 6-15:

Figure 6-15. Java output concerning the procedure of handling two conflicting calls.

The results in Figure 6-15 display the decision-making procedures for two calls, one for Yang's voice call to Anthony (mark <I.i>) and the other for Louisa's voice call to Anthony (mark <I.ii>). We regulate Session Keeper to check the communication statuses of all users involved in the suspended sessions every second. Therefore, for the scenario constructed in Figure 6-14, after having checked Anthony's status three times (marks <II.i>, <II.ii>, and <II.iii>), the Keeper identified that Anthony had changed from busy to idle and it then instructed the delivery of the second call to Anthony (mark <II.iv>). Figure 6-16 displays the recorded call executing results in the MySQL database.



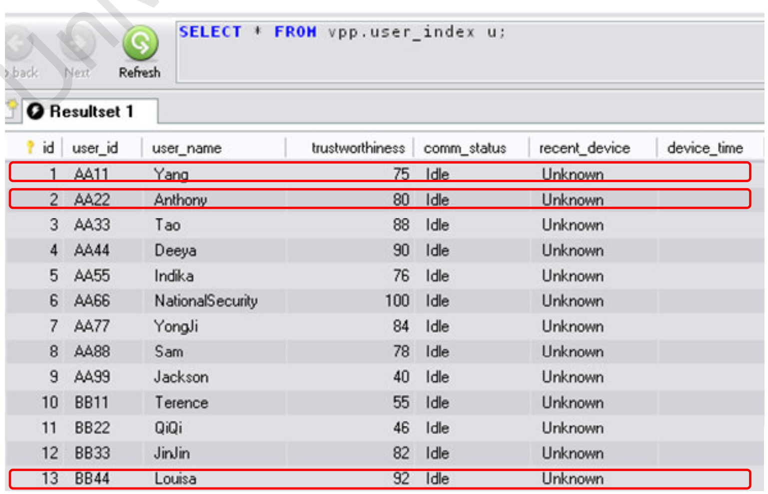
session_id	real_session_id	event_id	start_time	end_time	result	action_series
21C40D7D...	A1560F6383...	BB44 -> 0031 -> AA22 at 14:00:11	2007-08-10 14:00:11	2007-08-10 14:00:16	Success	-> Null -> Postpone(3) -> Force
CA0CB6C4...	5FD1259260...	AA11 -> 0031 -> AA22 at 14:00:01	2007-08-10 14:00:01	2007-08-10 14:00:03	Success	-> Null -> Deliver

Figure 6-16. MySQL results concerning the procedure of handling two conflicting calls.

The second row of data in Figure 6-16 records the first call from Yang (user ID “AA11”) to Anthony (user ID “AA22”) (mark <I.i>) and the first row records the second call from Louisa (user ID “BB44”) to Anthony (mark <I.ii>). According to the second-row record, it took the system two seconds to make the decision of successfully delivering the first call. According to the first-row record, the system forced the delivery of the second call after having postponed the call for two seconds (the system investigated the deliverability of the suspended call every second). In total, it took the system five seconds to make the decision of forcing the delivery of the call, including the two-second waiting period.

6.1.6.1 Users’ Real-time Statuses Changing with Session Status

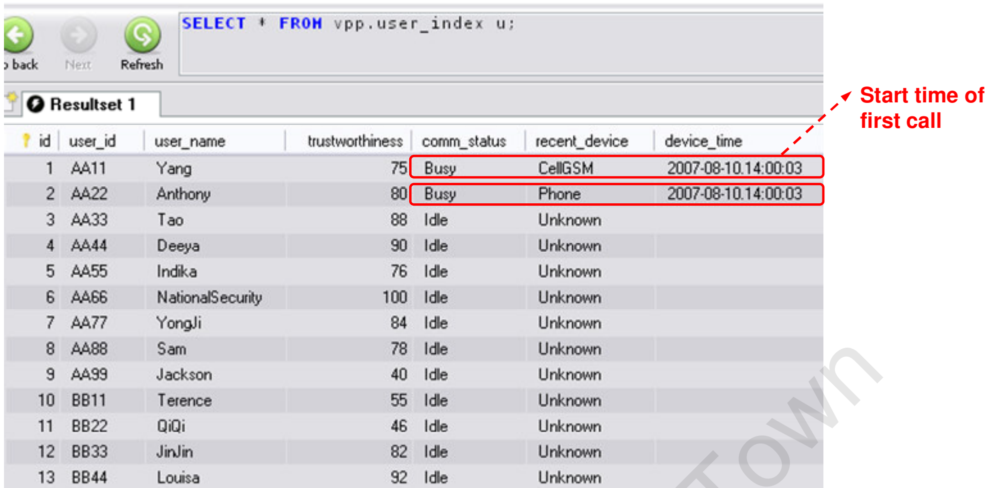
By tracking users’ real-time communication status (section 6.1.5.1), we can establish the stages of the sessions with which users are involved. Figure 6-17 to Figure 6-21 below exhibit the change of user status in the above communication scenario.



id	user_id	user_name	trustworthiness	comm_status	recent_device	device_time
1	AA11	Yang	75	Idle	Unknown	
2	AA22	Anthony	80	Idle	Unknown	
3	AA33	Tao	88	Idle	Unknown	
4	AA44	Deeya	90	Idle	Unknown	
5	AA55	Indika	76	Idle	Unknown	
6	AA66	NationalSecurity	100	Idle	Unknown	
7	AA77	YongJi	84	Idle	Unknown	
8	AA88	Sam	78	Idle	Unknown	
9	AA99	Jackson	40	Idle	Unknown	
10	BB11	Terence	55	Idle	Unknown	
11	BB22	QiQi	46	Idle	Unknown	
12	BB33	JinJin	82	Idle	Unknown	
13	BB44	Louisa	92	Idle	Unknown	

Figure 6-17. Stage 1 – Users were all idle before any call started.

The data records in Figure 6-17 show that Yang, Anthony, and Louisa were idle in terms of communication status before any call started.



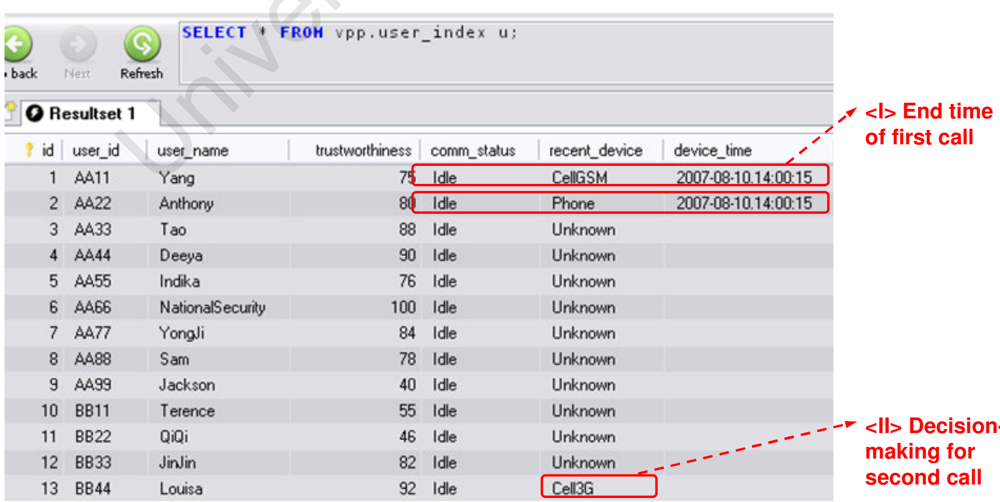
```
SELECT * FROM vpp.user_index u;
```

Resultset 1

id	user_id	user_name	trustworthiness	comm_status	recent_device	device_time
1	AA11	Yang	75	Busy	CellGSM	2007-08-10 14:00:03
2	AA22	Anthony	80	Busy	Phone	2007-08-10 14:00:03
3	AA33	Tao	88	Idle	Unknown	
4	AA44	Deeya	90	Idle	Unknown	
5	AA55	Indika	76	Idle	Unknown	
6	AA66	NationalSecurity	100	Idle	Unknown	
7	AA77	YongJi	84	Idle	Unknown	
8	AA88	Sam	78	Idle	Unknown	
9	AA99	Jackson	40	Idle	Unknown	
10	BB11	Terence	55	Idle	Unknown	
11	BB22	QiQi	46	Idle	Unknown	
12	BB33	JinJin	82	Idle	Unknown	
13	BB44	Louisa	92	Idle	Unknown	

Figure 6-18. Stage 2 – Users of the first call were busy during first-call execution.

After the system had instructed the physical network to execute the first call from Yang to Anthony at 14:00:03, these two users' statuses changed to busy as shown in Figure 6-18. Meanwhile, the database recorded their most recently used devices and their respective lasted used time. From 14:00:03, the statuses of the users are supposed to be continuously busy for 12 seconds according to system regulation.



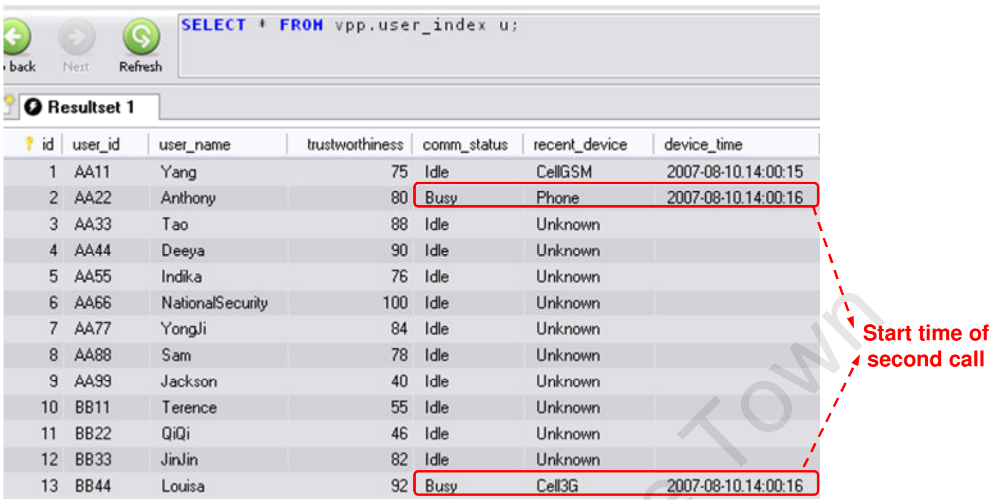
```
SELECT * FROM vpp.user_index u;
```

Resultset 1

id	user_id	user_name	trustworthiness	comm_status	recent_device	device_time
1	AA11	Yang	75	Idle	CellGSM	2007-08-10 14:00:15
2	AA22	Anthony	80	Idle	Phone	2007-08-10 14:00:15
3	AA33	Tao	88	Idle	Unknown	
4	AA44	Deeya	90	Idle	Unknown	
5	AA55	Indika	76	Idle	Unknown	
6	AA66	NationalSecurity	100	Idle	Unknown	
7	AA77	YongJi	84	Idle	Unknown	
8	AA88	Sam	78	Idle	Unknown	
9	AA99	Jackson	40	Idle	Unknown	
10	BB11	Terence	55	Idle	Unknown	
11	BB22	QiQi	46	Idle	Unknown	
12	BB33	JinJin	82	Idle	Unknown	
13	BB44	Louisa	92	Idle	Cell3G	

Figure 6-19. Stage 3 – Users of the first call became idle after first-call execution.

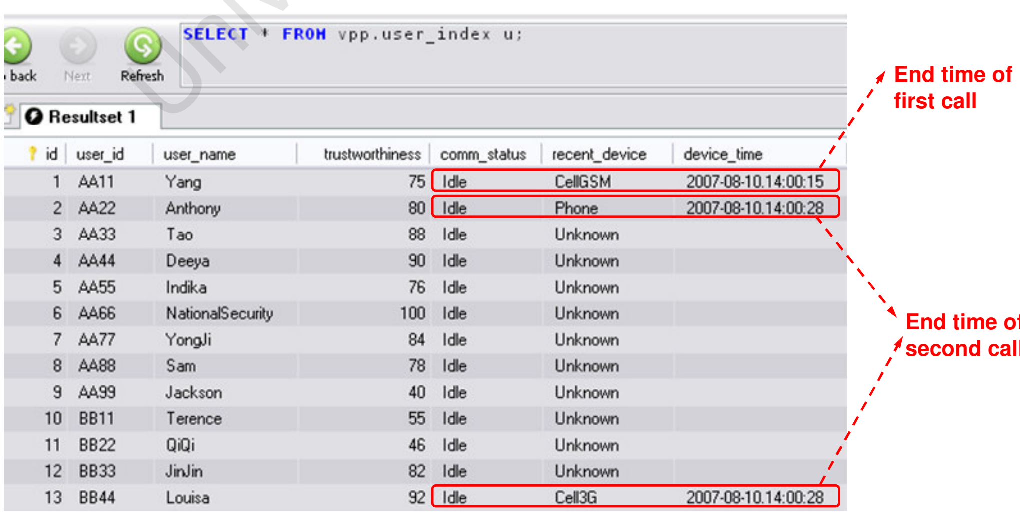
On the completion of the first call at 14:00:15, the statuses of both Yang and Anthony changed back to idle (mark <I> in Figure 6-19). Meanwhile, the system immediately re-evaluated the deliverability of Louisa's requested call to Anthony (mark <II>).



id	user_id	user_name	trustworthiness	comm_status	recent_device	device_time
1	AA11	Yang	75	Idle	CellGSM	2007-08-10 14:00:15
2	AA22	Anthony	80	Busy	Phone	2007-08-10 14:00:16
3	AA33	Tao	88	Idle	Unknown	
4	AA44	Deeya	90	Idle	Unknown	
5	AA55	Indika	76	Idle	Unknown	
6	AA66	NationalSecurity	100	Idle	Unknown	
7	AA77	YongJi	84	Idle	Unknown	
8	AA88	Sam	78	Idle	Unknown	
9	AA99	Jackson	40	Idle	Unknown	
10	BB11	Terence	55	Idle	Unknown	
11	BB22	QiQi	46	Idle	Unknown	
12	BB33	JinJin	82	Idle	Unknown	
13	BB44	Louisa	92	Busy	Cell3G	2007-08-10 14:00:16

Figure 6-20. Stage 4 – Users of the second call were busy during second-call execution.

Louisa's call to Anthony started executing at 14:00:16 and, from then on, the statuses of Louisa and Anthony became busy as shown in Figure 6-20. It is interesting to note that the latest time when Yang used her GSM cell phone was the time when her call with Anthony ended, whereas the latest time when Anthony and Louisa used their most recent devices was the time when their call started executing, for their call was still in process.



id	user_id	user_name	trustworthiness	comm_status	recent_device	device_time
1	AA11	Yang	75	Idle	CellGSM	2007-08-10 14:00:15
2	AA22	Anthony	80	Idle	Phone	2007-08-10 14:00:28
3	AA33	Tao	88	Idle	Unknown	
4	AA44	Deeya	90	Idle	Unknown	
5	AA55	Indika	76	Idle	Unknown	
6	AA66	NationalSecurity	100	Idle	Unknown	
7	AA77	YongJi	84	Idle	Unknown	
8	AA88	Sam	78	Idle	Unknown	
9	AA99	Jackson	40	Idle	Unknown	
10	BB11	Terence	55	Idle	Unknown	
11	BB22	QiQi	46	Idle	Unknown	
12	BB33	JinJin	82	Idle	Unknown	
13	BB44	Louisa	92	Idle	Cell3G	2007-08-10 14:00:28

Figure 6-21. Stage 5 – Users of the second call became idle after second-call execution.

After Louisa and Anthony finished the call, their communication statuses changed back to idle (Figure 6-21). Their most recently used devices and the corresponding time of using these devices were also updated.

6.2 Determining Session-Delivery Manners using Intelligence

In theory (sections 4.2.4.2 and 5.3.4), the virtual-user system has six manners to deliver a session. We illustrate these delivery manners in terms of functioning mechanism and system decision-making result under the communication scenario constructed in Table 6-2. The manners of *deliver* (section 6.2.1) and *fail* (section 6.2.2) belong to the conventional delivery that is applicable to both general- and intelligence-network, whereas those of *force* (section 6.2.3), *postpone* (section 6.2.4), *help* (section 6.2.5), and *learn* (section 6.2.6) belong to the remedial delivery that is only applicable to the intelligence network.

Table 6-2. Communication scenario for analysing the six session-delivery manners.

Variable	Unit	Content	Notes
session type	N/A	Call session, or file-sharing session	
traffic type	N/A	Real-time traffic	
device type	N/A	Phone, cell, or computer$^{\Delta}$	
session-execution start time	SystemTime	N/A	Generated as required
session connecting lifetime	second	Call session: 120	
		File-sharing session: 240	
duration of session execution	second	20	

* Refer to Appendix C for relevant user information in terms of personal detail, schedule, and social relationship.

$^{\Delta}$ Phone, cell, and computer individually provide the best-, good-, and normal-quality of service.

In Table 6-2, we purposely prolong the session connecting lifetime and reduce the duration of session execution to give prominence to the intelligence decision-making mechanism in the session connecting period. Hereof, the session connecting lifetime is the duration that the human-intelligence part takes to make decisions on the optimal delivery manner for a session. Yet the physical-network part factually executes the session in the session-execution duration.

Furthermore, we may get three combinations of decision-making results and final action for carrying out the above six session-delivery manners, as shown in Figure 6-22.

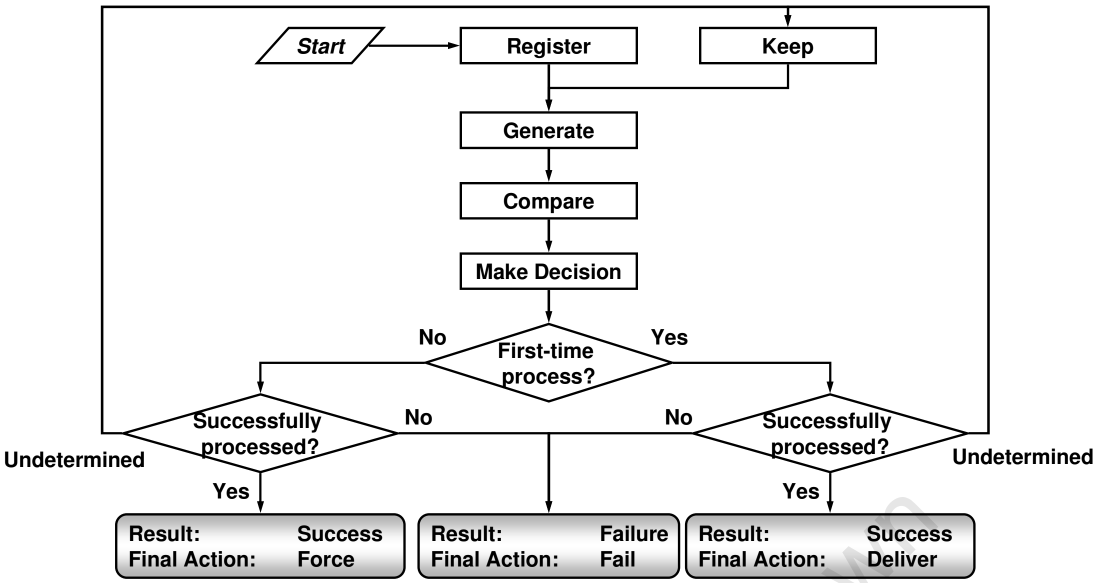


Figure 6-22. Categories of decision-making result with determined action.

Figure 6-22 illustrates the possible combinations of the decision-making result with the final determined action. If the system determines to successfully deliver a session after having only used the conventional manner (i.e., the system going through the procedure of generating, comparing, and decision-making for the session only once), the output result for the session will be "*Success*" and the final determined action will be "*Deliver*". If the system determines to deliver a session after having used any of the remedial manners (i.e., the system going through a procedure of postponing, generating, comparing, and decision-making), the output result will still be "*Success*" but the final action will be "*Force*". If the system finally determines to fail a session after having tried either the conventional- or the remedial-manner, the result will be "*Failure*" and the final action will be "*Fail*".

6.2.1 Conventional Manner – Deliver

The content from sections 6.1.1 to 6.1.5 has illustrated the successful delivery of a call session.

6.2.2 Conventional Manner – Fail

If the system determines to fail a session, it then does not need to try intelligence manners to save the session. For the communication case described in Table 6-3, the system had to fail the session proposed by the case, with the resulting Java output shown in Figure 6-23 and the MySQL record shown in Figure 6-24.

Table 6-3. Communication case for illustrating the manner of *fail*.

Initiator	Receiver	Application	Session Start Time	Expected Manner
Yang	Anthony	Phone-teleconferencing	2007-08-24.18:00:01	Fail

Table 6-3 provides the information to construct a scenario where Yang requested the start of a teleconferencing session to Anthony's office phone at 18:00:01 on 24 August 2007.

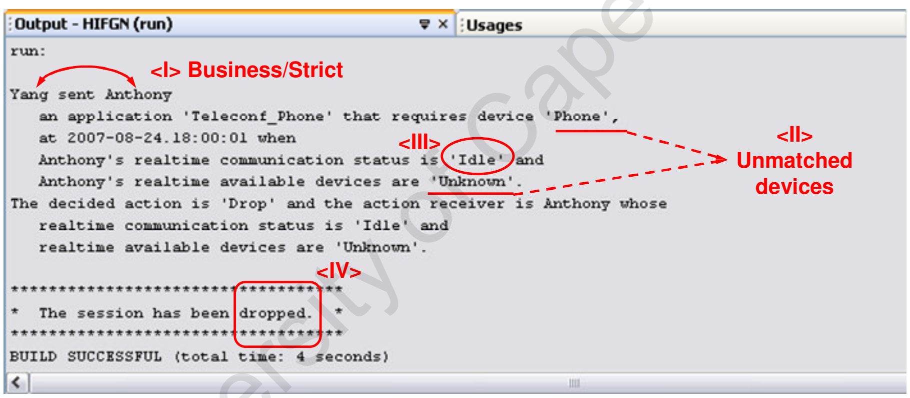


Figure 6-23. Java illustrated decision-making result for the manner of *fail*.

Figure 6-23 illustrates a session-delivery scenario where the manner of *fail* applies. Yang and Anthony are socially related in the "*business*" domain, so the strictness of their social relationship demanding on session delivery is "*strict*" (mark <I>). That is, the related two require neither a downgraded- nor a delayed-session. At 18:00:01 on 24 August 2007, Yang tried to initiate a phone-teleconference session to Anthony on his office phone. However, Anthony was at a "*public*" location where he was only available to some "*unknown*" device at the time. The system then had to fail Yang's requested session because Anthony was personally unavailable on the required phone (marks <II> and <IV>) and their strict relationship allowed no remedial measures for session delivery.

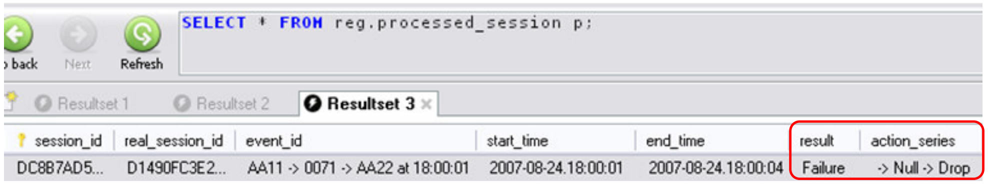


Figure 6-24. MySQL recorded decision-making result for the manner of *fail*.

The system recorded the final determined result in the database as shown in Figure 6-24 and, in reality, instructed the physical network to fail Yang's teleconferencing request.

6.2.3 Remedial Manner According to Session Property – Force

The system can force the delivery of a session to users either directly or indirectly. (1) Two types of session undertake *direct force*: one type comprises the sessions that can only provide poorer performance than expected and the other possesses an emergency feature. In what follows, the *force* action written in the format of "the manner of *force*" refers to *direct force*. (2) A forced delivery determined after any manner of *postpone*, *help*, or *learn* is an *indirect force*. In what follows, we refer to an *indirect force* by the manner that contributes the greatest to the final forced delivery. For example, the manner of forcing the delivery of a session after postponing it for a certain period will be referred to as the manner of *postpone* instead of the manner of *force*.

Based on the cases described in Table 6-4, we illustrate the manner of *force* with normal features and that with emergency features in terms of functioning mechanism and decision-making result respectively in Figure 6-25 and Figure 6-26. Figure 6-27 lists the records for the decision-making results of these sessions in the MySQL database.

Table 6-4. Communication cases for illustrating the manner of *force*.

Initiator	Receiver	Application	Start Time	Expected Manner	Feature of *force*
Yang	Sam	Call to phone	2007-08-24.18:00:01	Force	Normal
NationalSecurity	Yang	Call to cell	2007-08-24.18:00:10	Force	Emergency

Table 6-4 provides the information to construct a scenario where Yang used her cell phone to call Sam on his office phone at 18:00:01 on 24 August 2007 and, during their conversation, National Security called Yang to her cell.

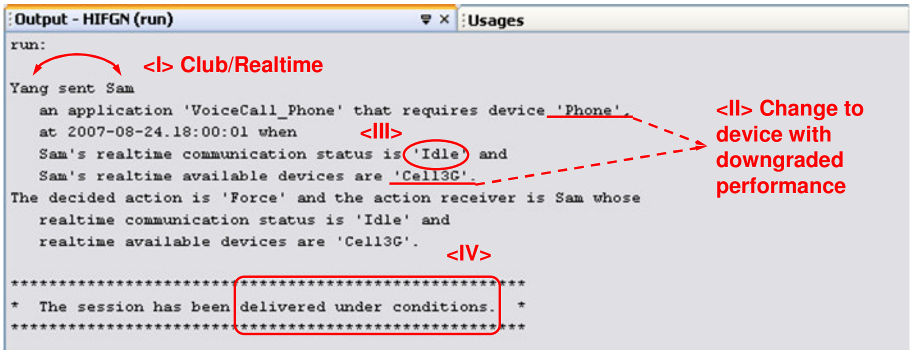


Figure 6-25. Java illustrated decision-making result for the manner of *normal force*.

Figure 6-25 illustrates a session-delivery scenario where the manner of *normal force* applies. Yang and Sam are socially related in the "*club*" domain and the strictness of their relationship requires a "*real-time*" session delivery between them (mark <I>). Yang initiated a voice call to Sam's office phone at 18:00:01, which Sam would not be able to answer because he was in the "*club*" domain and only had his 3G cell phone with him (mark <II>). Therefore, the system determined to deliver the session through the manner of *force* to Sam's 3G cell phone (mark <IV>). The manner ensured an immediate delivery of the call yet conveyed the service with poorer performance than that of the expected phone.

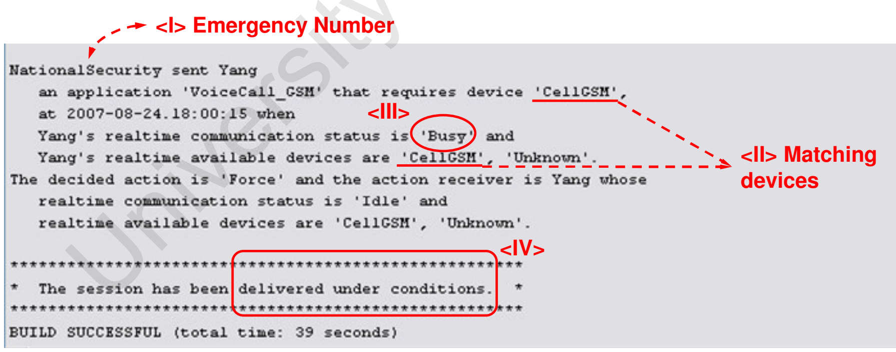
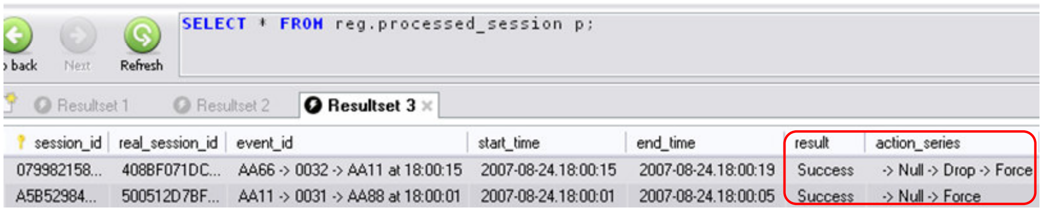


Figure 6-26. Java illustrated decision-making result for the manner of *emergency force*.

Figure 6-26 illustrates a session-delivery scenario where the manner of *emergency force* applies. During the conversation of Sam and Yang, National Security urgently called Yang on her cell (mark <I>). Although Yang was available with the right device – her GSM cell phone (mark <II>), she could not answer the call because she was personally busy with a call to Sam (mark <III>). However, the call from National Security was an emergency that required urgent delivery.

The system then had to break Yang’s on-going call session with Sam and force the delivery of the urgent call from National Security to Yang (mark <IV>).



session_id	real_session_id	event_id	start_time	end_time	result	action_series
079982158...	408BF071DC...	AA66 -> 0032 -> AA11 at 18:00:15	2007-08-24 18:00:15	2007-08-24 18:00:19	Success	-> Null -> Drop -> Force
A5B52984...	500512D7BF...	AA11 -> 0031 -> AA88 at 18:00:01	2007-08-24 18:00:01	2007-08-24 18:00:05	Success	-> Null -> Force

Figure 6-27. MySQL recorded decision-making result for the manner of *force*.

Figure 6-27 lists the two delivered sessions in the above scenario. The system delivered the first call (application ID “0031”) from Yang (user ID “AA11”) to Sam (user ID “AA88”) with a downgraded quality of service (the second row in Figure 6-27). It delivered the second call (application ID “0032”) from National Security (user ID “AA66”) to Yang due to the emergency characteristic of the call (the first row in Figure 6-27). The manner of *emergency force* saved a would-be-failed session by making use of its emergency feature.

6.2.4 Remedial Manner According to Session Property – Postpone

The system considers postponing a session when the session is involved in either of the following two communication situations. One is where the expected receiver is immediately available but the required device will only become available in future. The other is where the required device is immediately available but the expected receiver is busy at that moment. We call the probability that the expected receiver succeeds in receiving the session in the first situation the receiver’s *current availability* and that in the second situation the receiver’s *scheduled availability*.

Based on the cases described in Table 6-5, we illustrate the decision-making results for the manner of *postpone* when the receiver is currently available and when the receiver will be available according to its schedule respectively in Figure 6-28 and Figure 6-29. Figure 6-30 lists the decision-making results of these sessions in the MySQL database.

Table 6-5. Communication cases for illustrating the manner of *postpone*.

Initiator	Receiver	Application	Start Time	Expected Manner	Feature of *postpone*
National Security	Yang	Call to cell	2007-08-24.19:59:01	Deliver	N/A
JinJin	Yang	Call to cell	2007-08-24.19:59:20	Postpone	Current availability
QiQi	Yang	Video file-sharing	2007-08-24.19:59:55	Postpone	Scheduled availability

Table 6-5 provides the information to construct a scenario where three communication sessions occurred. Yang received a call on her cell phone from National Security at 19:59:01 on 24 August 2007 and, during their conversation, JinJin called Yang on her cell. At 19:59:55, QiQi started sharing a video file with Yang via her computer.

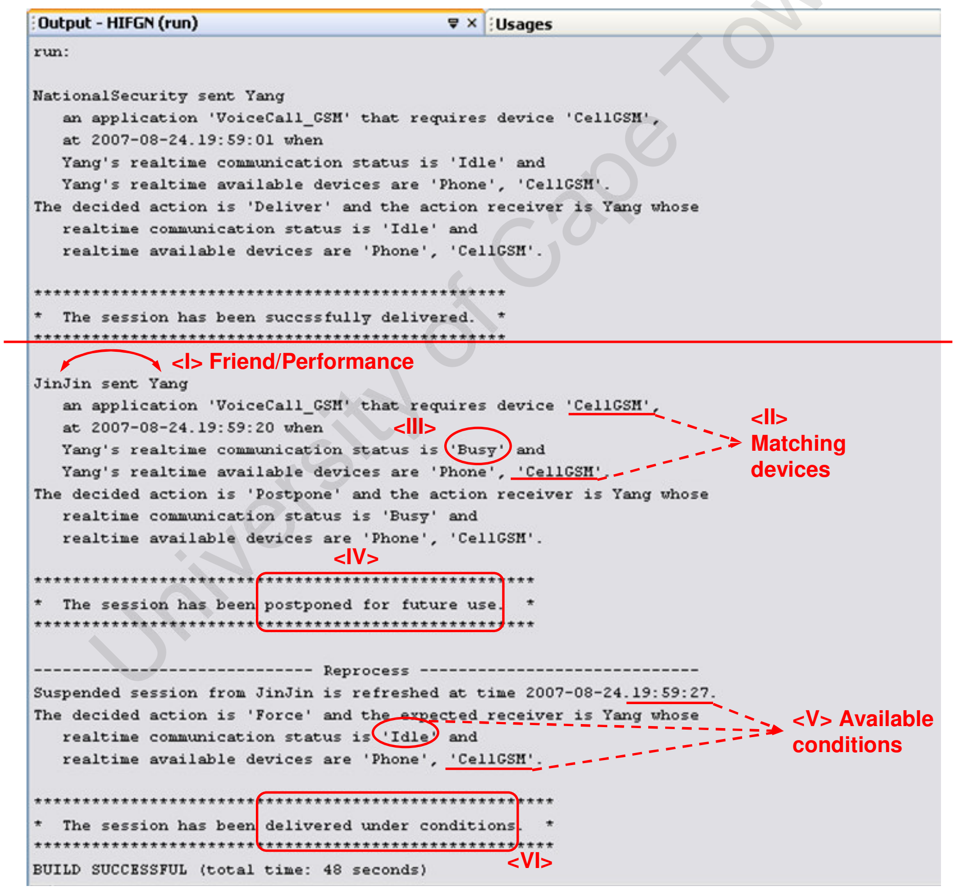


Figure 6-28. Java illustrated decision-making result for the manner of *current postpone*.

Figure 6-28 illustrates a session-delivery scenario where the manner of *current postpone* applies. National Security successfully called Yang at 19:59:01 and Yang's communication status became busy for 20 seconds thereafter. When JinJin called Yang at 19:59:20, Yang was still busy with the call from National Security. Although Yang was with the required device – her GSM cell phone (mark <II>), she was unavailable due to her "*busy*" communication status (mark <III>). Because JinJin and Yang knew each other in the "*friend*" domain where they had an "*easy*" relationship (mark <I>), the system determined to postpone the requested call from JinJin for a second attempt in future (mark <IV>). In this scenario, JinJin only needed to wait for a short while (maximum the entire length of National Security's call to Yang) until Yang's communication status changed to idle.

At 19:59:27 when Yang became available on her GSM cell (mark <V>), the system instructed the delivery of JinJin's requested call to Yang (mark <VI>). Until then, the call had been waiting in the system for seven seconds, which is quite acceptable for a user to wait for to connect a normal call. The call would be over at 19:59:47 thereafter Yang's communication status became idle.

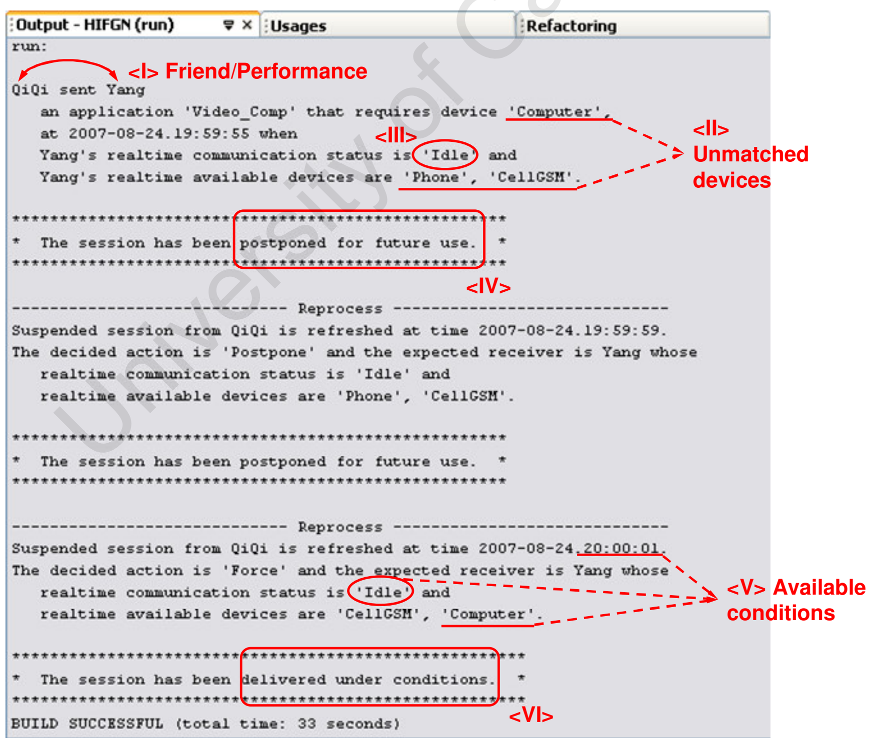
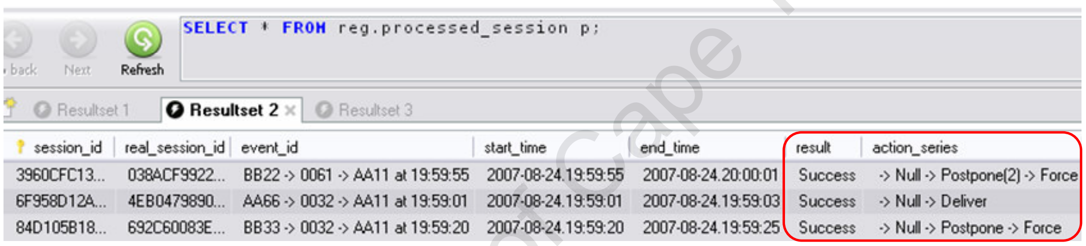


Figure 6-29. Java illustrated decision-making result for the manner of *scheduled postpone*.

Figure 6-29 illustrates a session-delivery scenario where the manner of *scheduled postpone* applies. QiQi requested to share with Yang a video file on her computer at 19:59:55. Yang was free at that time yet not available on the proper device – her computer (marks <II> and <III>). The system then planned to suspend the session in the database for 240 seconds, which represents the connecting life for a video-sharing-on-computer service. Fortunately, Yang was back at the "*office*" location at 20:00:00 according to her schedule and became available for the "*computer*" from that time (mark <V>). Then QiQi's request for a file-sharing service was deliverable to Yang at 20:00:00 (mark <VI>). In this case, the requested session only waited in the system for five seconds. In reality, a determined-to-be-postponed session can wait for a long time for a successful delivery, only if it is still within the session connecting lifetime. For example, a data service such as Email may wait for 24 hours to deliver.



session_id	real_session_id	event_id	start_time	end_time	result	action_series
3960CFC13...	038ACF9922...	BB22 -> 0061 -> AA11 at 19:59:55	2007-08-24 19:59:55	2007-08-24 20:00:01	Success	-> Null -> Postpone(2) -> Force
6F958D12A...	4EB0479890...	AA66 -> 0032 -> AA11 at 19:59:01	2007-08-24 19:59:01	2007-08-24 19:59:03	Success	-> Null -> Deliver
84D105B18...	692C60083E...	BB33 -> 0032 -> AA11 at 19:59:20	2007-08-24 19:59:20	2007-08-24 19:59:25	Success	-> Null -> Postpone -> Force

Figure 6-30. MySQL recorded the decision-making result for the manner of *postpone*.

Figure 6-30 lists the three delivered sessions in the above scenario, two of which were first postponed and then delivered. The first call (application ID "0032") from National Security (user ID "AA66") to Yang (user ID "AA11") was successfully delivered thereafter Yang was busy from 19:59:03 to 19:59:23 (the second row in Figure 6-30). The second call (application ID "0032") from JinJin (user ID "BB33") to Yang started processing at 19:59:20, where the system identified that Yang was busy at the time. The call then waited until 19:59:23 when Yang became available. After re-evaluating the deliverability at 19:59:25, the system finally delivered the second call that kept Yang busy for another 20 seconds until 19:59:45 (the third row in Figure 6-30). The file-sharing session (application ID "0061") from QiQi (user ID "BB22") to Yang started after the second call had finished but it still needed to wait for Yang to become available for the proper device according to her schedule. It therefore only started executing at 20:00:01 (the first row in Figure 6-30). We call the way to postpone the second call session according to

Yang's real-time communication status "the manner of *current postpone*" and that to postpone the file-sharing session according to Yang's communication schedule "the manner of *scheduled postpone*".

6.2.5 Remedial Manner According to Social Knowledge – Help

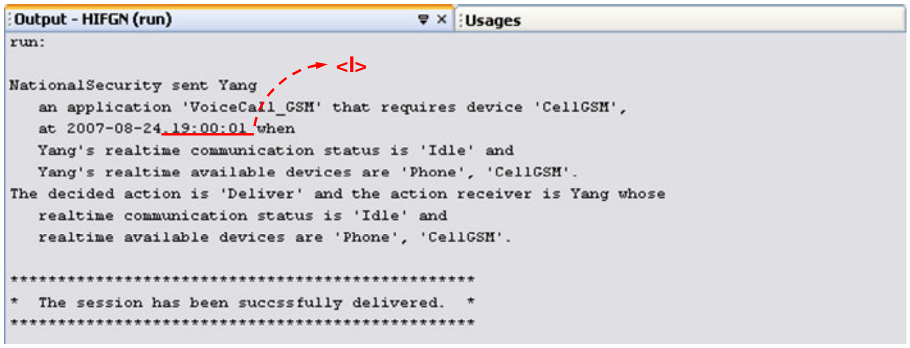
The system performs the manner of *help* when needed and to whom to appeal for aid is up to the social relationship of the originally expected session involvers. The help that happens as a session requires is called "the manner of *senior help*" and the help that happens after the manner of *current postpone* fails to facilitate the delivery is called "the manner of *subordinate help*".

Based on the cases described in Table 6-6, we illustrate the decision-making results for the manner of *senior help* and that of *subordinate help* respectively in Figure 6-32 and Figure 6-33 when the original expected receiver was involved in the session depicted in Figure 6-31. Figure 6-34 lists the decision-making results of these sessions in the MySQL database.

Table 6-6. Communication cases for illustrating the manner of *help*.

Initiator	Receiver	Application	Start Time	Expected Manner	Expected Helper
NationalSecurity	Yang	Call to cell	2007-08-24.19:00:01	Deliver	N/A
Indika	Yang	Call to cell	2007-08-24.19:00:10	Help	Terence
YongJi	Yang	Video file-sharing to computer	2007-08-24.19:00:11	Help	Tao

Table 6-6 provides the information to construct a scenario where the following three communication sessions occurred. Yang received a call on her cell phone from National Security at 19:00:01 on 24 August 2007. During their conversation, Indika called Yang to her cell and YongJi started sharing a video file with Yang on her computer.



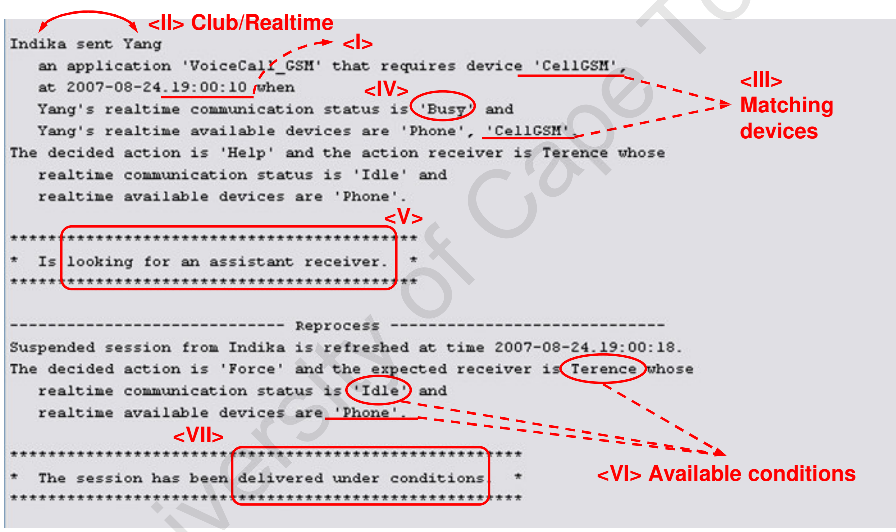
```
Output - HIFGN (run)                    Usages
run:

NationalSecurity sent Yang
   an application 'VoiceCall_GSM' that requires device 'CellGSM',
   at 2007-08-24 19:00:01 when
   Yang's realtime communication status is 'Idle' and
   Yang's realtime available devices are 'Phone', 'CellGSM'.
The decided action is 'Deliver' and the action receiver is Yang whose
   realtime communication status is 'Idle' and
   realtime available devices are 'Phone', 'CellGSM'.

*************************************************
*  The session has been succssfully delivered.  *
*************************************************
```


Figure 6-31. Java illustrated decision-making result for preparing for a *help* manner.

In Figure 6-31, National Security called Yang at 19:00:01 (mark <I>) and Yang became busy until 19:00:21, if excluding the decision-making period.



```
Indika sent Yang
   an application 'VoiceCall_GSM' that requires device 'CellGSM',
   at 2007-08-24 19:00:10 when
   Yang's realtime communication status is 'Busy' and
   Yang's realtime available devices are 'Phone', 'CellGSM'.
The decided action is 'Help' and the action receiver is Terence whose
   realtime communication status is 'Idle' and
   realtime available devices are 'Phone'.

*****************************************
*  Is looking for an assistant receiver.  *
*****************************************

---------------------------- Reprocess ----------------------------
Suspended session from Indika is refreshed at time 2007-08-24 19:00:18.
The decided action is 'Force' and the expected receiver is Terence whose
   realtime communication status is 'Idle' and
   realtime available devices are 'Phone'.

*************************************************
*  The session has been delivered under conditions  *
*************************************************
```


Figure 6-32. Java illustrated decision-making result for the manner of *senior help*.

Figure 6-32 illustrates a session-delivery scenario where the manner of *senior help* applies. Indika called Yang to her GSM cell phone at 19:00:10 (marks <I> and <III>). However, Yang could not receive the call because she was still busy with the call from National Security (mark <IV>). Indika and Yang know each other in the "*club*" domain where all sessions require "*real-time*" delivery (mark <II>), thus the system needs to urgently look for a third trusted user who is capable of receiving the call for Yang (mark <V>). Abiding by the rules (sections 4.2.4.2 and 5.3.6), the system identified a qualified helper Terence who was in the common domain with Indika and Yang and was immediately available on some proper device (mark <VI>). The system thus instructed Terence to help Yang with the call (mark <VII>). This type of *senior help* facilitates the session delivery by following the first suggestion from the system.

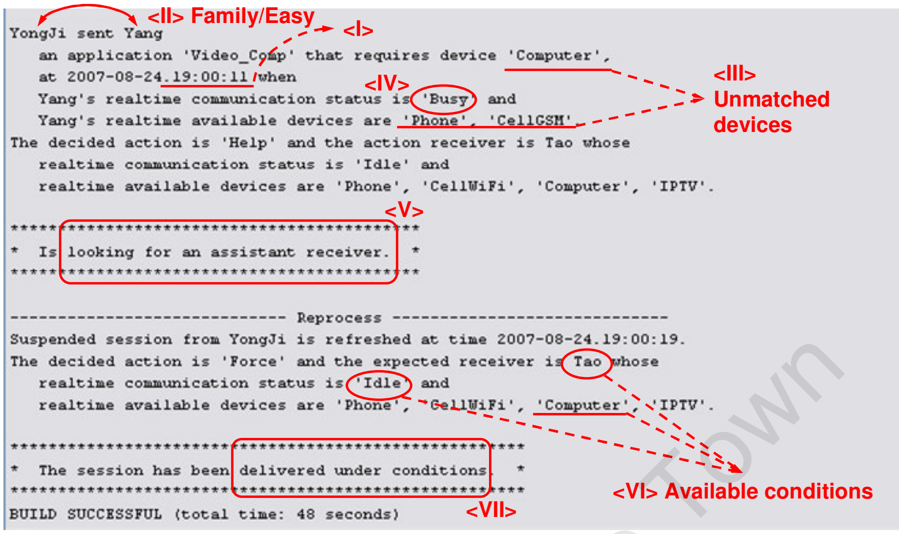
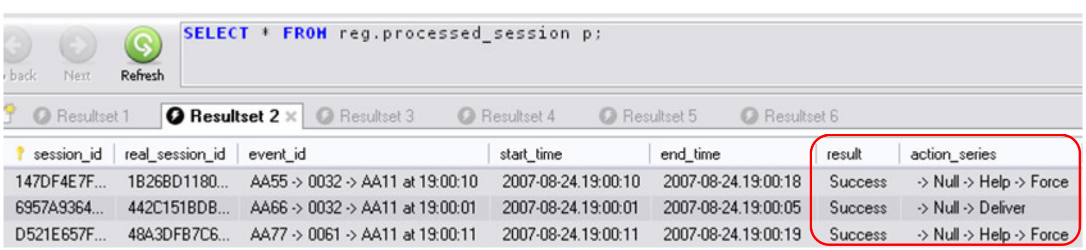


Figure 6-33. Java illustrated decision-making result for the manner of *subordinate help*.

Figure 6-33 illustrates a session-delivery scenario where the manner of *subordinate help* applies. YongJi also requested to share a video file with Yang when Yang was busy (mark <IV>). Because YongJi and Yang are socially related in the "*family*" domain (mark <II>), their "*easy*" relationship allows the system to first try the manner of *postpone* and, if unsuccessful, then try the manner of *help* (mark <V>). Based on Yang's communication status and schedule, she would not become available for the file-sharing session within the session connecting lifetime, so the system eventually determined to use the manner of *help* to facilitate session delivery (mark <VII>). The final identified helper was Tao, who related to both Yang and YongJi in the "*family*" domain and was real-time available on the computer at the decision-making moment (mark <VI>).



session_id	real_session_id	event_id	start_time	end_time	result	action_series
147DF4E7F...	1B26BD1180...	AA55 -> 0032 -> AA11 at 19:00:10	2007-08-24.19:00:10	2007-08-24.19:00:18	Success	-> Null -> Help -> Force
6957A9364...	442C151BDB...	AA66 -> 0032 -> AA11 at 19:00:01	2007-08-24.19:00:01	2007-08-24.19:00:05	Success	-> Null -> Deliver
D521E657F...	48A3DFB7C6...	AA77 -> 0061 -> AA11 at 19:00:11	2007-08-24.19:00:11	2007-08-24.19:00:19	Success	-> Null -> Help -> Force

Figure 6-34. MySQL recorded decision-making result for the manner of *help*.

Figure 6-34 lists three delivered sessions in the above scenario, two of which the system had to use the manner of *help* to deliver. The first successfully delivered call (application ID "0032") from National Security (user ID "AA66") to Yang (user ID "AA11") required Yang's communication status to be busy from 19:00:05 to 19:00:25 (the second row in Figure 6-34). During the executing period of the first call, Indika (user ID "AA55") called Yang at 19:00:10 and this call expected an immediate delivery. The system therefore identified a qualified assistant receiver – Terence – to receive the call for Yang (the first row in Figure 6-34). Also during this period, YongJi (user ID "AA77") wanted to send Yang a video file at 19:00:11. After having found that Yang would not be able to receive the session even after suspending the session for a long-enough time, the system finally identified a trustworthy user – Tao – to help Yang receive the file (the third row in Figure 6-34).

6.2.6 Remedial Manner According to Social Knowledge – Learn

If the initiator of a session is not in the social-relationship list of the receiver, or the receiver is uncertain how to determine the delivery manner for the session, the receiver has to learn from his/her commonly related users with the initiator on how to deliver the session.

Based on the cases described in Table 6-7, Figure 6-35 illustrates the decision-making results for the manner of *learn*. Figure 6-36 lists the decision-making results of these sessions in the MySQL database.

Table 6-7. Communication cases for illustrating the manner of *learn*.

Initiator	Receiver	Application	Start Time	Expected Manner
QiQi	Tao	Call to computer	2007-08-24.20:01:01	Learn

Table 6-7 provides the information to construct a scenario where QiQi would like to call Tao on his computer at 20:01:01 on 24 August 2007.

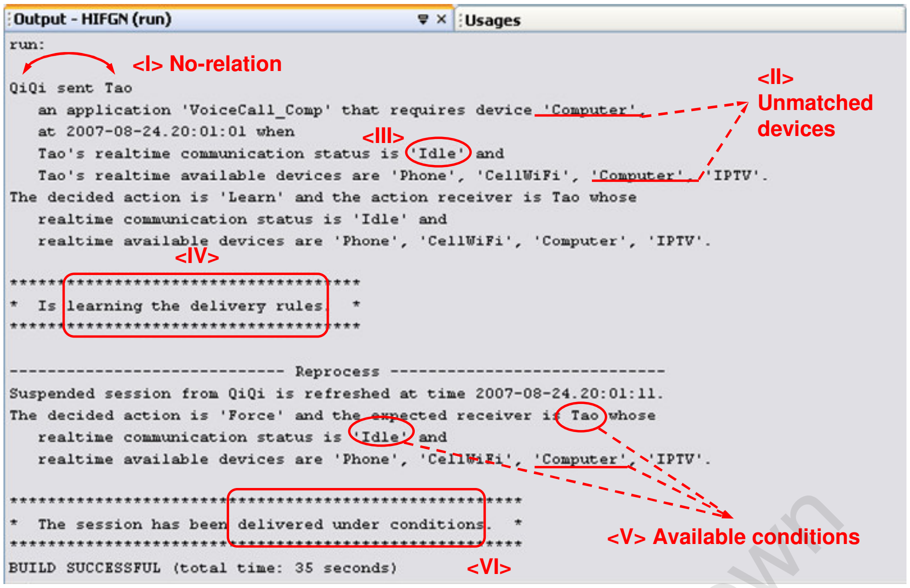


Figure 6-35. Java illustrated decision-making result for the manner of *learn*.

In Figure 6-35, QiQi attempted to call Tao on his computer when Tao was real-time available at his computer (marks <II> and <III>). However, because QiQi and Tao were not in each other's contact lists (mark <I>), the system instructed QiQi to first learn what to do with the session and then determined the final delivery (mark <IV>). Based on the learning rules (section 5.3.7), the system succeeded in seeking out a social-relationship connection between QiQi and Tao via their commonly related users. That is, virtual QiQi and Tao may succeed in establishing a social relationship between them. The system finally instructed the physical network to connect QiQi's requested call to Tao who had been proven qualified to receive it (mark <VI>).

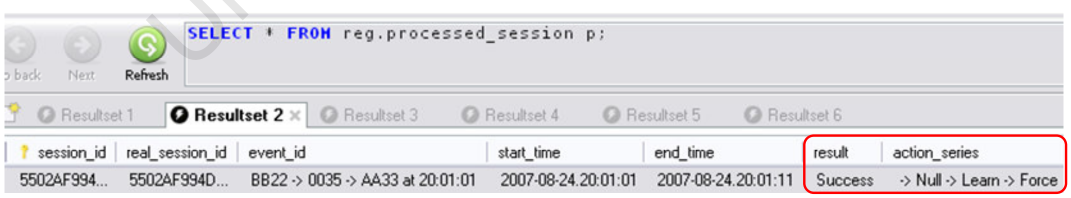


Figure 6-36. MySQL recorded decision-making result for the manner of *learn*.

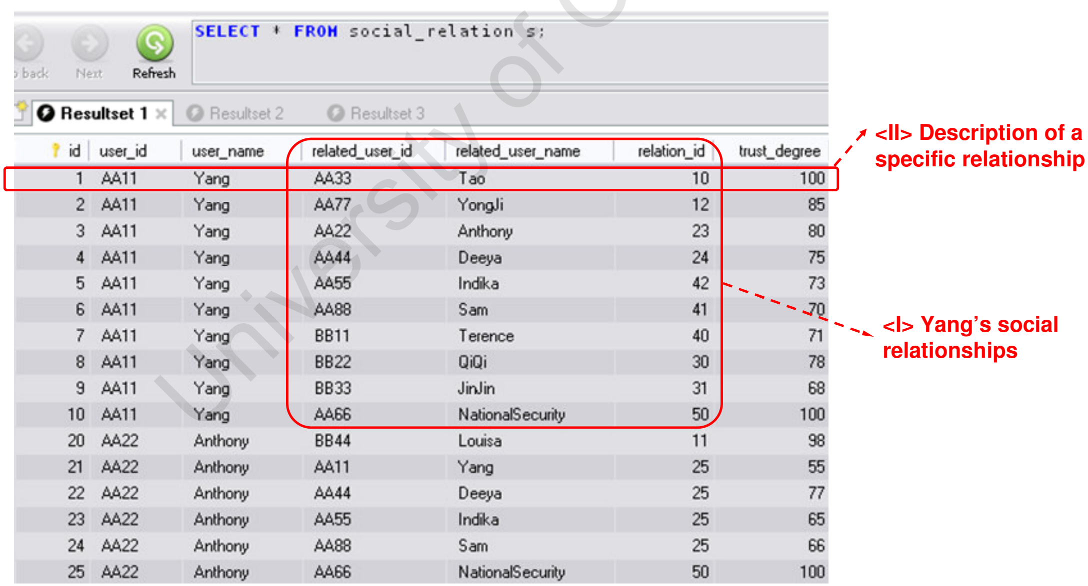
Figure 6-36 lists the delivered session in the above scenario. The system eventually forced the delivery of the call (application ID "0035") requested by QiQi (user ID "BB22") to Tao (user ID "AA33") after having performed the learning mechanism.

6.3 Representing Social Knowledge

To help the network determine optimal session delivery, the system adds in several social features that work on behalf of human-like intelligence. The first feature is quantifying trust degree, which embodies any two users' abstract social relationship (section 6.3.1). The second is collaborating between the users' absolute trustworthiness values and their trust-degree values (section 6.3.2). The third is using a user's schedule to determine his/her real-time communication availability at a given moment (section 6.3.3).

6.3.1 Embodiment of Two Users' Social Relationship

The virtual-user system mainly maintains two tables to accommodate users' social-relationship features. One table contains all social connections for each network user (Figure 6-37) and the other describes each type of social connection using relevant information (Figure 6-38).

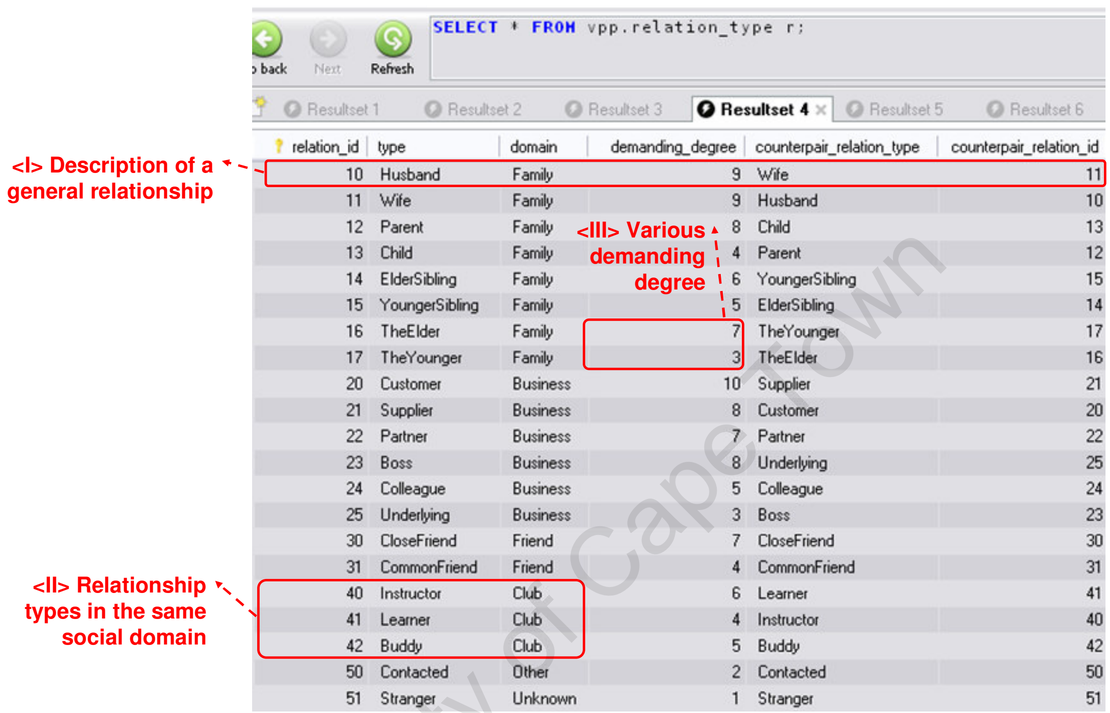


```
SELECT * FROM social_relation s;
```

id	user_id	user_name	related_user_id	related_user_name	relation_id	trust_degree
1	AA11	Yang	AA33	Tao	10	100
2	AA11	Yang	AA77	YongJi	12	85
3	AA11	Yang	AA22	Anthony	23	80
4	AA11	Yang	AA44	Deeya	24	75
5	AA11	Yang	AA55	Indika	42	73
6	AA11	Yang	AA88	Sam	41	70
7	AA11	Yang	BB11	Terence	40	71
8	AA11	Yang	BB22	QiQi	30	78
9	AA11	Yang	BB33	JinJin	31	68
10	AA11	Yang	AA66	NationalSecurity	50	100
20	AA22	Anthony	BB44	Louisa	11	98
21	AA22	Anthony	AA11	Yang	25	55
22	AA22	Anthony	AA44	Deeya	25	77
23	AA22	Anthony	AA55	Indika	25	65
24	AA22	Anthony	AA88	Sam	25	66
25	AA22	Anthony	AA66	NationalSecurity	50	100

Figure 6-37. A fragment of social-relationship table in MySQL database.

Figure 6-37 illustrates the format of the "social_relation" table where the system stores network users' social relationships. Yang (user ID "AA11") has 10 socially related users in her contact list, with each relating to her in an exclusive relationship type (mark <I>). The record for the relationship of Yang to another user is composed of Yang's user ID and user name, the related user's ID and name, the ID of their relationship type, and the degree that Yang trusts the related user in receiving her sessions (mark <II>). The relationship type conforms to a fixed format as shown in Figure 6-38.



relation_id	type	domain	demanding_degree	counterpair_relation_type	counterpair_relation_id
10	Husband	Family	9	Wife	11
11	Wife	Family	9	Husband	10
12	Parent	Family	8	Child	13
13	Child	Family	4	Parent	12
14	ElderSibling	Family	6	YoungerSibling	15
15	YoungerSibling	Family	5	ElderSibling	14
16	TheElder	Family	7	TheYounger	17
17	TheYounger	Family	3	TheElder	16
20	Customer	Business	10	Supplier	21
21	Supplier	Business	8	Customer	20
22	Partner	Business	7	Partner	22
23	Boss	Business	8	Underlying	25
24	Colleague	Business	5	Colleague	24
25	Underlying	Business	3	Boss	23
30	CloseFriend	Friend	7	CloseFriend	30
31	CommonFriend	Friend	4	CommonFriend	31
40	Instructor	Club	6	Learner	41
41	Learner	Club	4	Instructor	40
42	Buddy	Club	5	Buddy	42
50	Contacted	Other	2	Contacted	50
51	Stranger	Unknown	1	Stranger	51

Figure 6-38. Social-relationship-type table in MySQL database.

Figure 6-38 lists all social-relationship types used in the HIFGN project. The record for each piece of social-relationship information includes the relationship type, its situated social domain, the demanding degree, and the counterpair-relationship type. (1) The relationship-type item describes the relationship of the related user to the principal user. For example, if Yang is the principal user, Tao showing up in Yang's social-relationship list with relationship ID "10" (mark <II> in Figure 6-37) means that Tao (the related user) is Yang's husband (mark <I> in Figure 6-38). (2) The social domain confines the two related users within a specific social group. For example, if Terence relates to Yang as an "*instructor*" and Indika relates to Yang as a "*buddy*", then most probably Terence relates to Indika as an "*instructor*" as well and these three users participate in the same social activity in the "*club*" domain (mark <II>). The social-domain parameter assists in the manner of *learn* when the system performs intelligence decision-making (section 5.3.7). (3) The demanding degree with the value ranging from 0 to 10, signifies the urgency of the principal user requesting its related user to accomplish its session. The degree of eagerness greatly affects the acceleration of the session in priority if the session is in Session Keeper. The higher the degree of urgency, a better chance of succeeding a session stands. For example, if Yang gets two calls respectively from an older and a younger family member at the same time, the system by default directs the elder's call to Yang and fails the younger one's because the older member has a higher demanding degree ("7") than that of the younger ("3") (mark <III>). (4) The counterpair relationship is the counterpart of the observing relationship. It describes the relationship the other way around when compared with the observing one and is reserved for future use.

6.3.1.1 Social Relationship Affecting Session Delivery

Based on the cases described in Table 6-8, Figure 6-39 presents in detail how the social relationship of the session involvers affects the decisions made for session delivery. Figure 6-40 and Figure 6-41 individually list the decision-making result of the successfully delivered sessions and that of the suspended ones in the MySQL database.

Table 6-8. Cases for exhibiting the effect of social relationship on session delivery.

Initiator	Receiver	Initiator's Relation with Receiver	Social Domain	Application	Start Time	Expected Manner	Expected Assistant
Deeya	Yang	Colleague	Business	Call to cell	2007-08-21.08:00:01	Deliver	N/A
Anthony	Yang	Boss	Business	Call to cell	2007-08-21.08:00:11	Fail	N/A
Tao	Yang	Husband	Family	Call to cell	2007-08-21.08:00:11	Postpone	N/A
QiQi	Yang	CommonFriend	Friend	Call to cell	2007-08-21.08:00:11	Postpone	N/A
Sam	Yang	Learner	Club	Call to cell	2007-08-21.08:00:11	Help	Terence
Jackson	Yang	Stranger	Unknown	Call to cell	2007-08-21.08:00:11	Learn	Terence

Table 6-8 provides the information to construct a scenario where, when Yang is in a call conversation with Deeya, other users in different social domains with Yang called Yang to her cell at the same time. These users include Anthony, Tao, QiQi, Sam, and Jackson.

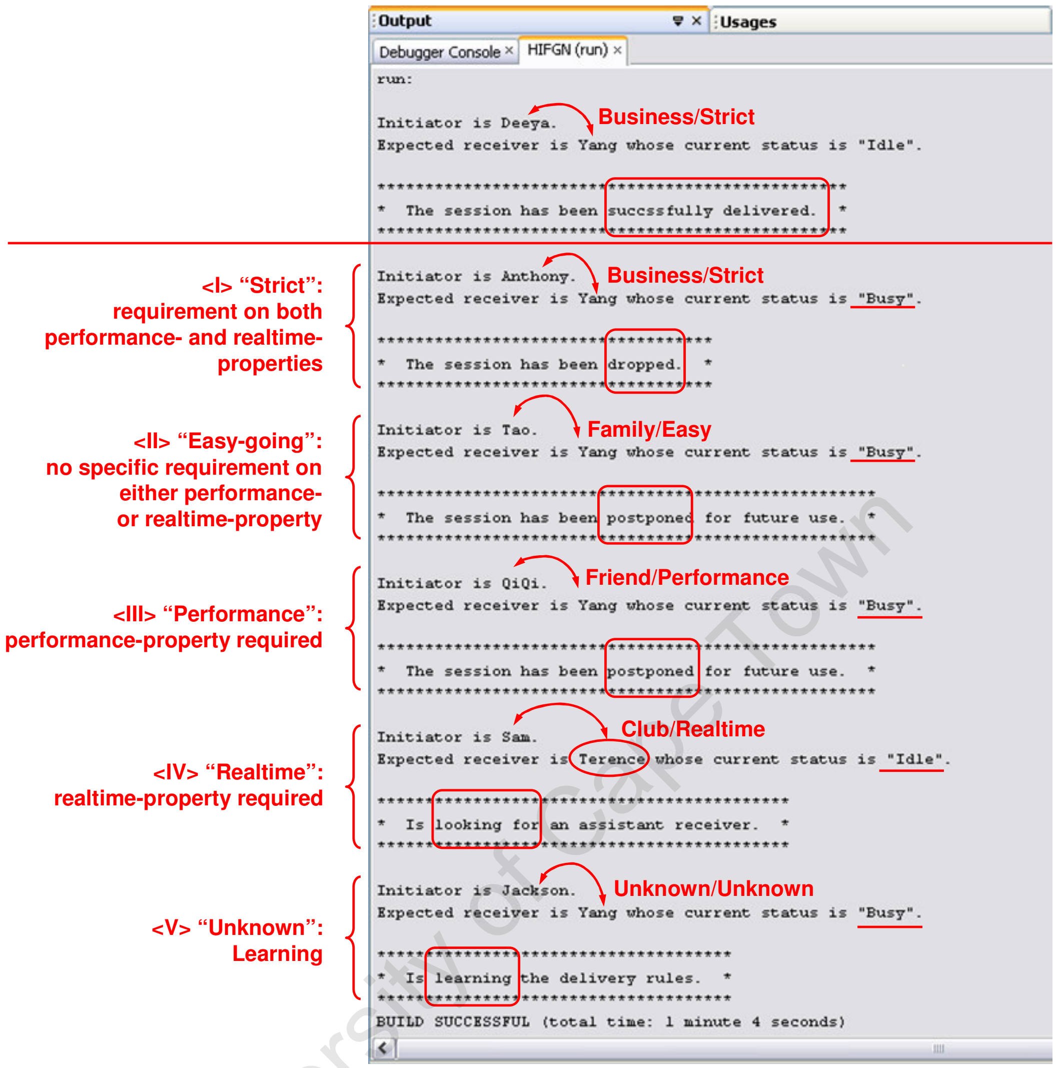


Figure 6-39. Effect of user social relationships on the determination of session delivery.

Figure 6-39 illustrates how users' relationship type plays an extraordinary role in decision-making about the most appropriate session delivery. After having answered Deeya's call, Yang's communication status changed to busy. Meanwhile, Anthony, Tao, QiQi, Sam, and Jackson called Yang. Because Anthony and Yang are socially related in the "*business*" domain where a "*strict*" requirement on session delivery applies, the system had to fail Anthony's call (mark <I>). Because Tao and Yang are socially related in the "*family*" domain where an "*easy*" requirement on session delivery applies, the system found that Yang would shortly be available and thus indicated postponing Tao's call (mark <II>). QiQi and Yang are socially related in the "*friend*" domain where a "*performance*" requirement on session delivery applies; the system thus suggested postponing the session until Yang became available to receive a quality-acceptable session (mark <III>). Because Sam and Yang are socially related in the "*club*" domain where a "*real-time*" requirement on session delivery applies, the system had to immediately seek a third trustable, qualified, and available user – Terence – to assist Yang with the call (mark <IV>). Jackson and Yang are related in the "*unknown*" domain with no record of relationship in each other's contact list (mark <V>). The system then established that they both relate to Terence in their respective relationships and then instructed to force the session.

After the above decision-makings, the system immediately failed Anthony's call (Figure 6-40) and suspended the rest in the network for further processing (Figure 6-41).

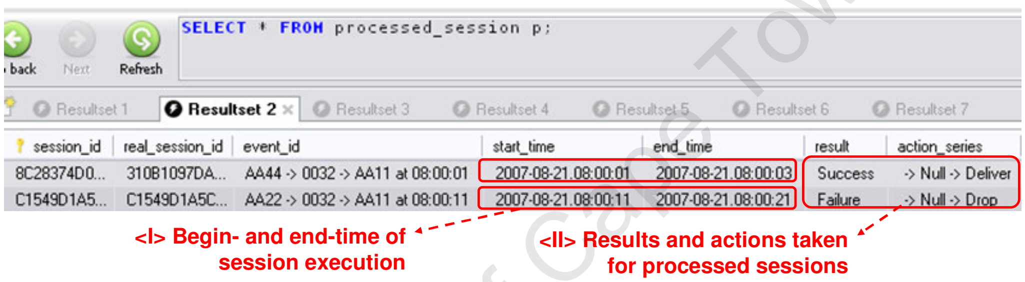


Figure 6-40. Immediately processed sessions for the cases in Table 6-8.

Figure 6-40 lists the two immediately processed sessions: the one contained in the first row is the call from Deeya (user ID "AA44") and that in the second row is the call from Anthony (user ID "AA22"). The database records the decision-making duration, the potential result, and the determined action towards the result for each session (marks <I> and <II>).

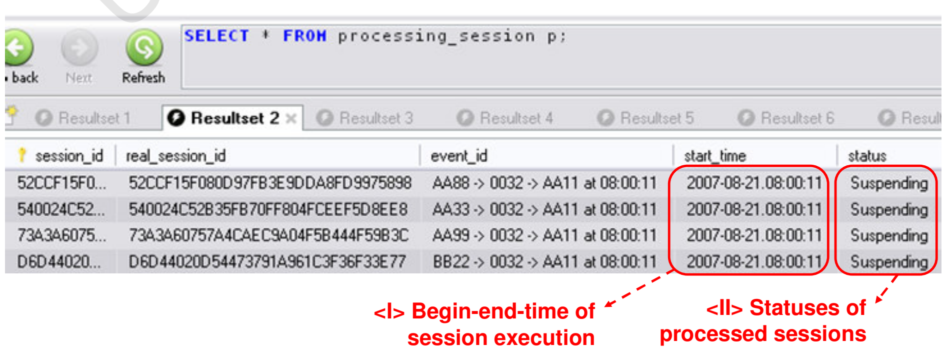


Figure 6-41. Suspended sessions for the cases in Table 6-8.

Figure 6-41 contains the four sessions that need reprocessing with the change of communication environment. These sessions include the requested calls from Tao (user ID "AA33"), QiQi (user ID "BB22"), Sam (user ID "AA88"), and Jackson (user ID "AA99"). The system saved these would-be-failed sessions using intelligence manoeuvres instead of immediately failing them as the general network does.

6.3.2 Cooperation of Absolute Trustworthiness and Trust Degree

In theory (section 5.3.6), the absolute trustworthiness and the trust degree collaborate to determine the potential qualified assistant for the scenario where the system uses the manner of *learn* or *help* to facilitate session delivery.

Based on the cases described in Table 6-9 and the users' trust-related features defined in Table 6-10 and Table 6-11, Figure 6-42 and Figure 6-43 together exhibit how users' absolute trustworthiness values and their inter-trust degrees collaboratively influence the determination of the most appropriate assistant user.

Table 6-9. Cases for illustrating the cooperation of trustworthiness and trust degree.

Initiator	Receiver	Social Domain	Application	Start Time	Expected Manner	Expected Assistant
Deeya	Yang	Business	Call to cell	2007-08-21.08:00:01	Deliver	N/A
Sam	Yang	Club	Call to cell	2007-08-21.08:00:11	Help	Terence, or Indika

Table 6-9 provides the information to construct a scenario where Sam called Yang to her cell when Yang is in a call conversation with Deeya. Because the session from Sam to Yang required an immediate delivery and Yang was too busy to receive it, the system had to look for an assistant user to receive the call for Yang. The expected assistant should be trustable, personally available, and technically qualified for the session. The following analysis focuses the trust-related issues.

Table 6-10. Absolute trustworthiness values of involved users and domain in Table 6-9.

User	Trustworthiness	Note
Yang	75	
Terence	55	For Figure 6-42.

	45	For Figure 6-43.
Indika	65	
“Club” domain	50	

Table 6-10 lists the absolute trustworthiness values of all related users and that of the “*club*” domain in the scenario in Table 6-9. The higher the trustworthiness value, the more the system recommends the user receive a communication session.

Table 6-11. Trust-degree values of the initiator to all potential receivers in Table 6-9.

Initiator	Trust degree of Sam to Yang	Trust degree of Sam to Terence	Trust degree of Sam to Indika
Sam	60	98	65

Table 6-11 lists the trust degree of Sam to all other potential receivers in the scenario in Table 6-9. The higher the trust-degree value, the more Sam personally trusts the related user in receiving a communication session.

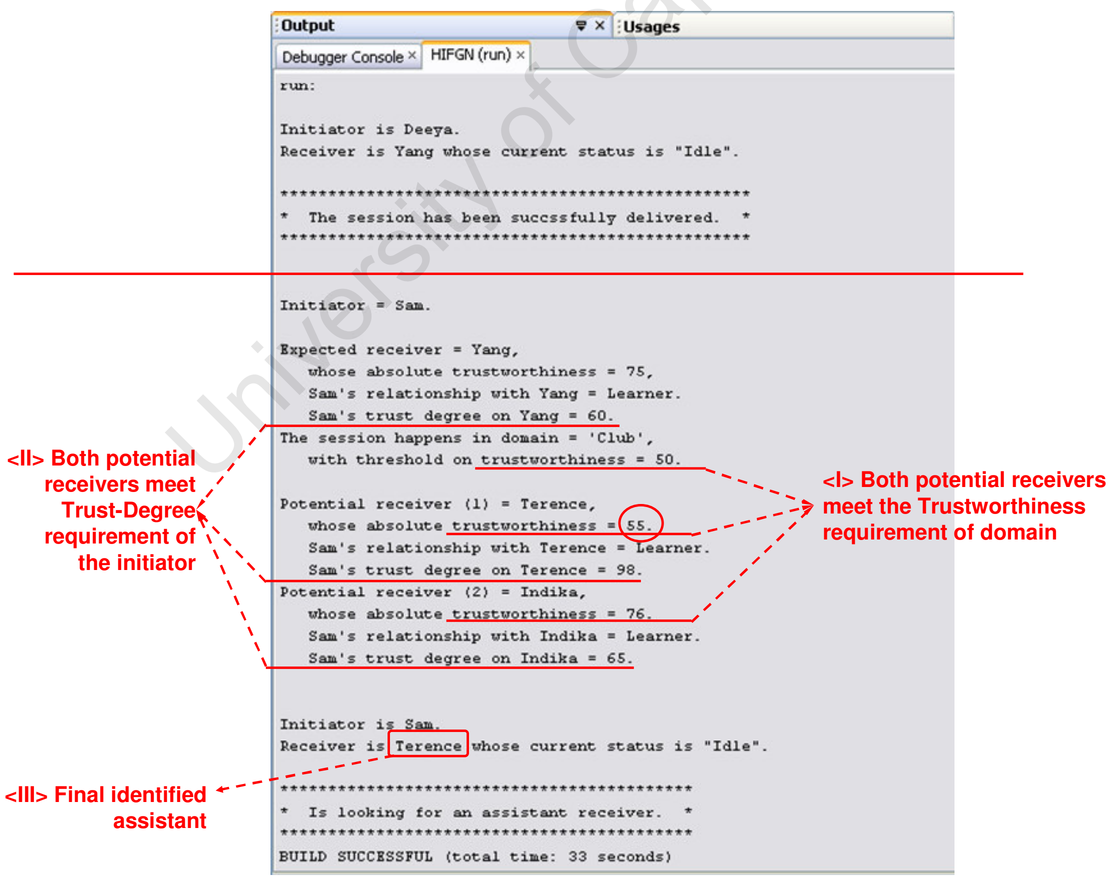


Figure 6-42. Example scenario where the trust-degree value determines session delivery.

Figure 6-42 illustrates the decision-making results for the cases in Table 6-9 when Terence's absolute trustworthiness value is "55". Because both Terence's trustworthiness value ("55") and Indika's trustworthiness value ("76") are greater than the minimum required trustworthiness of the "*club*" domain ("50") (mark <I>), the two users are therefore qualified to assist Sam with the session in terms of trustworthiness. Nevertheless, although Sam trusted Terence (with a high trust-degree value of "98") more than Indika (with a low trust-degree value of "65") (mark <II>), the system eventually selected Terence to be the assistant receiver and delivered Sam's call to Terence (mark <III>). This selection abides by the initiator's first preference.

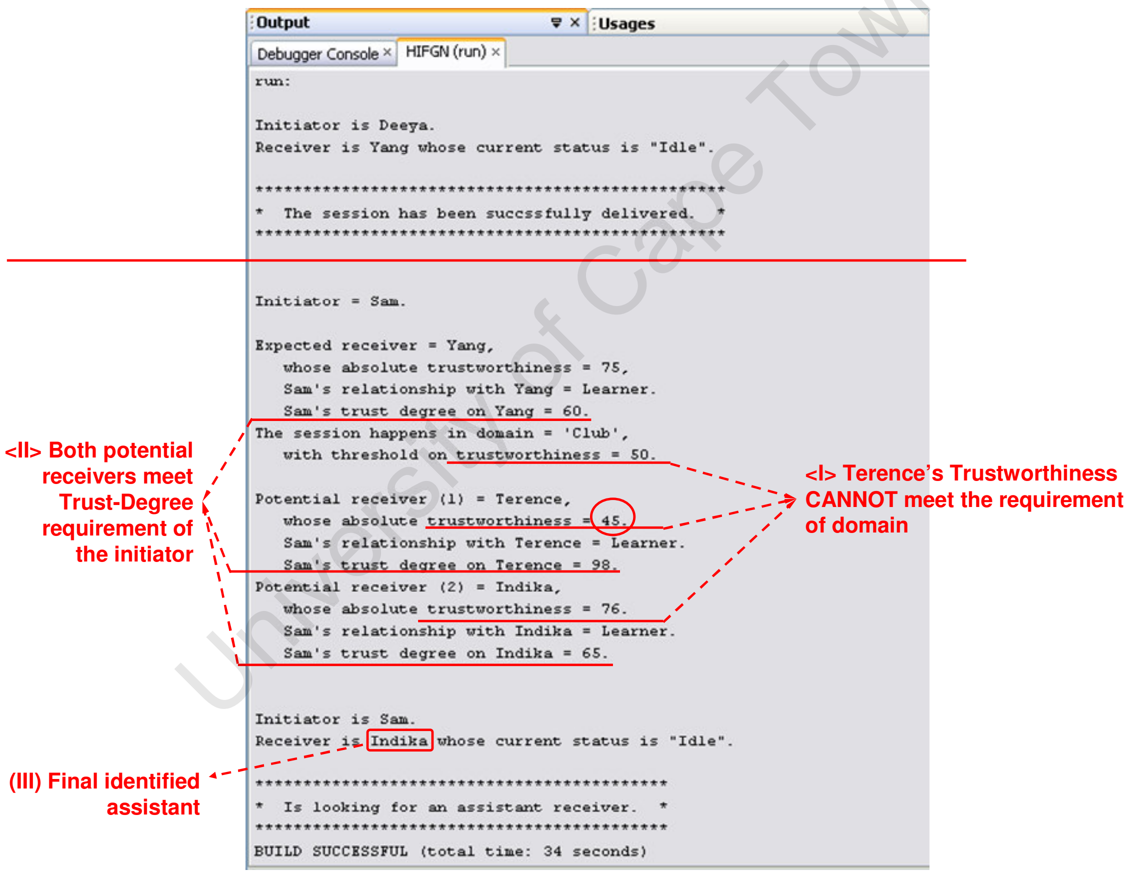


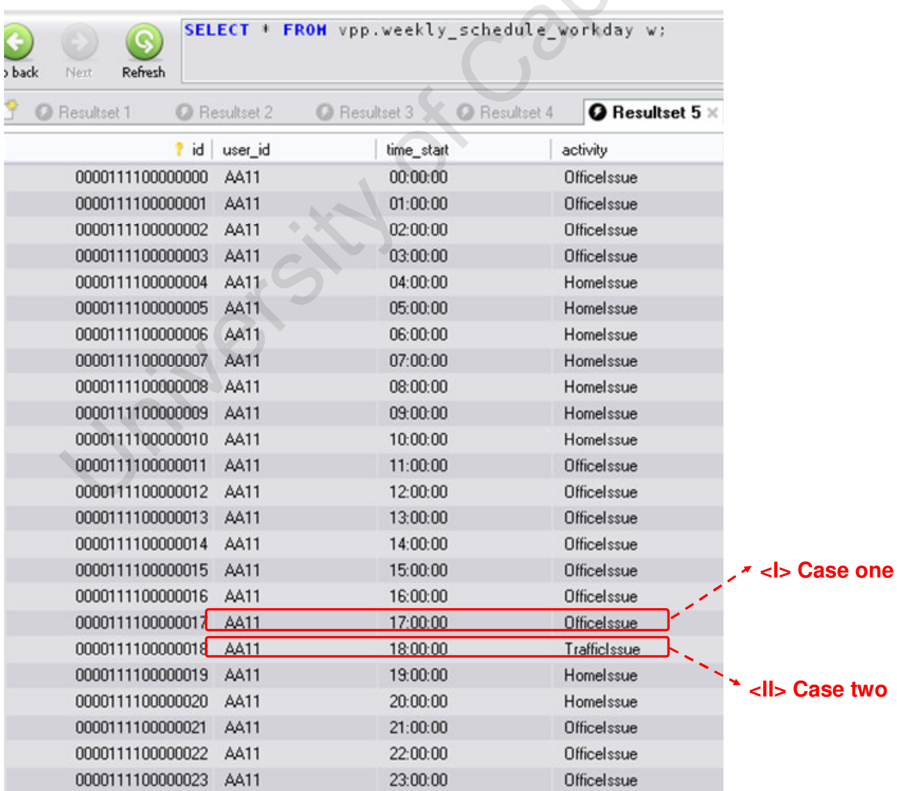
Figure 6-43. Example scenario where the trustworthiness value determines session delivery.

Figure 6-43 illustrates the decision-making results for the cases in Table 6-9 when Terence's absolute trustworthiness is "45", which does not meet the minimum trustworthiness requirement in the "*club*" domain ("50") (mark <I>). Therefore, no matter how much Sam trusted Terence in receiving the call for him (mark <II>), the system refused to use Terence as the final assistant. Instead, it would use Indika who was more trustable in the eyes of the system, although less trusted by the original initiator (mark <III>).

6.3.3 User's Schedule Determining Communication Availability

The user database stores all users' communication activities as well as their schedules. Only when necessary, does the system read the relevant information from the database into Virtual Personal Profile to facilitate decision-making.

According to the theory (section 5.3.1), the system uses the start time of a session to identify the involved users' communication availability, including real-time status and available devices. The sequential Figure 6-44 to Figure 6-46 illustrate such a procedure of using a user's schedule to determine its physical availability for communications at a given moment.

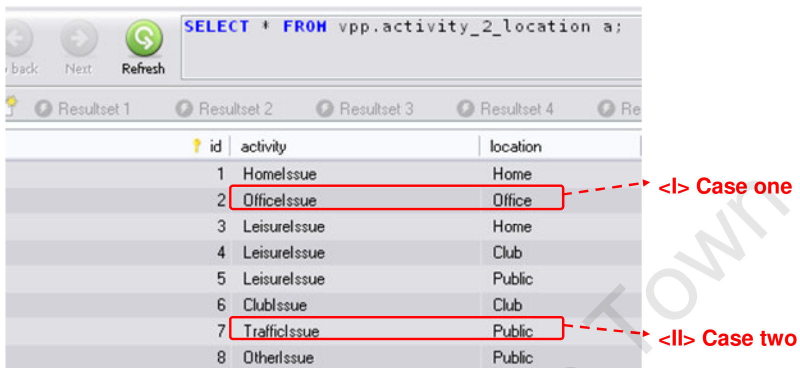


```
SELECT * FROM vpp.weekly_schedule_workday w;
```

id	user_id	time_start	activity
0000111100000000	AA11	00:00:00	OfficeIssue
0000111100000001	AA11	01:00:00	OfficeIssue
0000111100000002	AA11	02:00:00	OfficeIssue
0000111100000003	AA11	03:00:00	OfficeIssue
0000111100000004	AA11	04:00:00	HomeIssue
0000111100000005	AA11	05:00:00	HomeIssue
0000111100000006	AA11	06:00:00	HomeIssue
0000111100000007	AA11	07:00:00	HomeIssue
0000111100000008	AA11	08:00:00	HomeIssue
0000111100000009	AA11	09:00:00	HomeIssue
0000111100000010	AA11	10:00:00	HomeIssue
0000111100000011	AA11	11:00:00	OfficeIssue
0000111100000012	AA11	12:00:00	OfficeIssue
0000111100000013	AA11	13:00:00	OfficeIssue
0000111100000014	AA11	14:00:00	OfficeIssue
0000111100000015	AA11	15:00:00	OfficeIssue
0000111100000016	AA11	16:00:00	OfficeIssue
0000111100000017	AA11	17:00:00	OfficeIssue
0000111100000018	AA11	18:00:00	TrafficIssue
0000111100000019	AA11	19:00:00	HomeIssue
0000111100000020	AA11	20:00:00	HomeIssue
0000111100000021	AA11	21:00:00	OfficeIssue
0000111100000022	AA11	22:00:00	OfficeIssue
0000111100000023	AA11	23:00:00	OfficeIssue

Figure 6-44. Stage 1 – Identify user real-time activity at an instant.

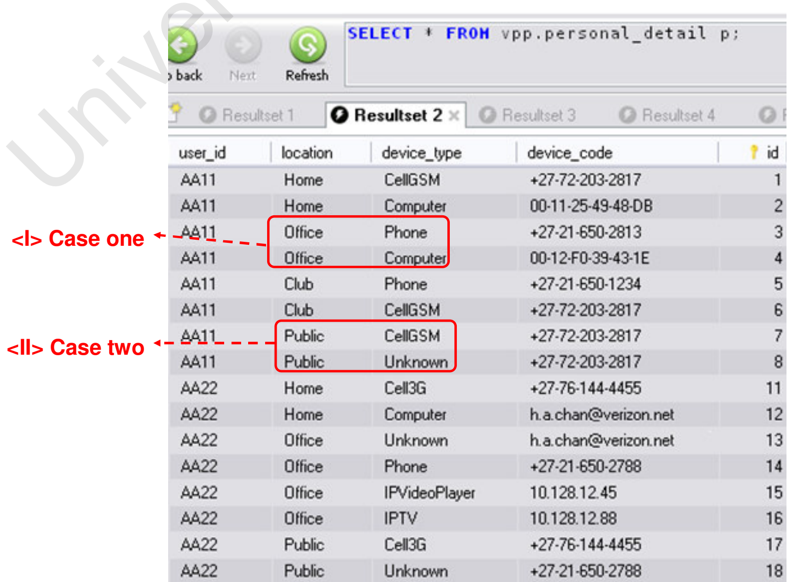
Figure 6-44 displays a fragment of a user's workday communication schedule. For simplicity, the system uses one-hour timeslots and defines a user's social activities in hours. For example, Yang (user ID "AA11") is normally busy with her "*office issue*" from 17:00:00 to 17:59:59 (mark <I>) and involved with her "*traffic issue*" from 18:00:00 to 18:59:59 (mark <II>) on workdays.



id	activity	location
1	HomeIssue	Home
2	OfficeIssue	Office
3	LeisureIssue	Home
4	LeisureIssue	Club
5	LeisureIssue	Public
6	ClubIssue	Club
7	TrafficIssue	Public
8	OtherIssue	Public

Figure 6-45. Stage 2 – Identify user real-time location according to user on-going activity.

Figure 6-45 displays the table in which the system defines the users' common social activities and the respective physical locations where these activities most likely occur. For example, an "*office issue*" generally happens in the "*office*" (mark <I>) and a "*traffic issue*" happens in the "*public*" (mark <II>).



user_id	location	device_type	device_code	id
AA11	Home	CellGSM	+27-72-203-2817	1
AA11	Home	Computer	00-11-25-49-48-DB	2
AA11	Office	Phone	+27-21-650-2813	3
AA11	Office	Computer	00-12-F0-39-43-1E	4
AA11	Club	Phone	+27-21-650-1234	5
AA11	Club	CellGSM	+27-72-203-2817	6
AA11	Public	CellGSM	+27-72-203-2817	7
AA11	Public	Unknown	+27-72-203-2817	8
AA22	Home	Cell3G	+27-76-144-4455	11
AA22	Home	Computer	h.a.chan@verizon.net	12
AA22	Office	Unknown	h.a.chan@verizon.net	13
AA22	Office	Phone	+27-21-650-2788	14
AA22	Office	IPVideoPlayer	10.128.12.45	15
AA22	Office	IPTV	10.128.12.88	16
AA22	Public	Cell3G	+27-76-144-4455	17
AA22	Public	Unknown	+27-21-650-2788	18

Figure 6-46. Stage 3 – User location determines real-time available devices.

Figure 6-46 exemplifies the association of a user's physical locations with their respective potential available devices. For example, Yang (user ID "AA11") is most likely available for her "*phone*" and "*computer*" when she is in her "*office*" (mark <I>) and available for her GSM "*cell*" in the "*public*" domain (mark <II>).

After having gone through the entire procedure, the system has learnt that Yang is generally available on her office phone and computer from 17:00:00 to 17:59:59 and available on her cell phone from 18:00:00 to 18:59:59 according to her preset communication schedule.

6.3.3.1 Example of Schedule Determining Communication Availability

Based on the cases described in Table 6-12, Figure 6-47 to Figure 6-49 present the effect of a user's present schedule on communication availability from different points of view.

Table 6-12. Cases for exhibiting the effect of user schedule on user availability.

Initiator	Receiver	Initiator's relation with Receiver	Social Domain	Application	Start Time	Expected Manner
Deeya	Yang	Colleague	Business	Call to phone	2007-08-23.17:59:01	Deliver
Deeya	Yang	Colleague	Business	Call to phone	2007-08-23:18:00:01	Fail

Table 6-8 provides the information to construct a scenario where Deeya first phoned Yang to her office phone at 17:59:01 and then again at 18:00:01.

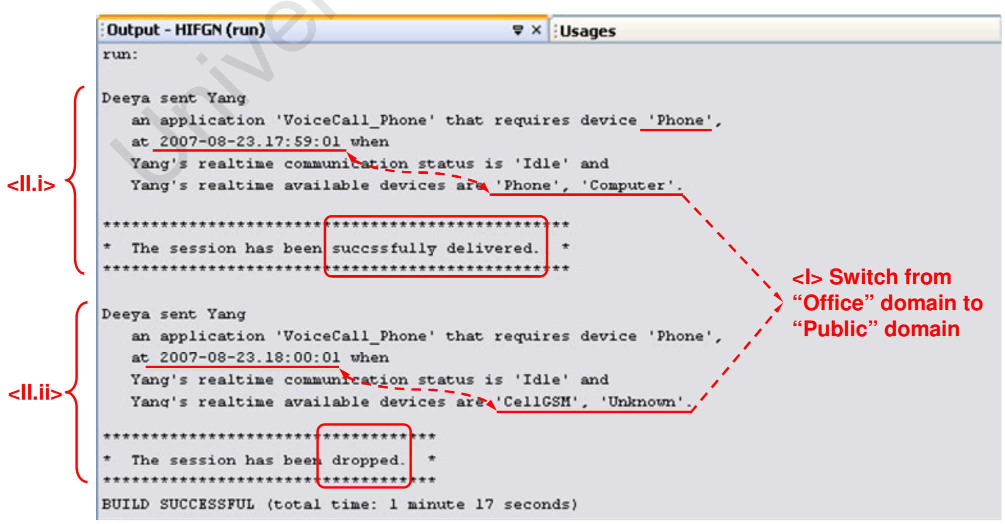
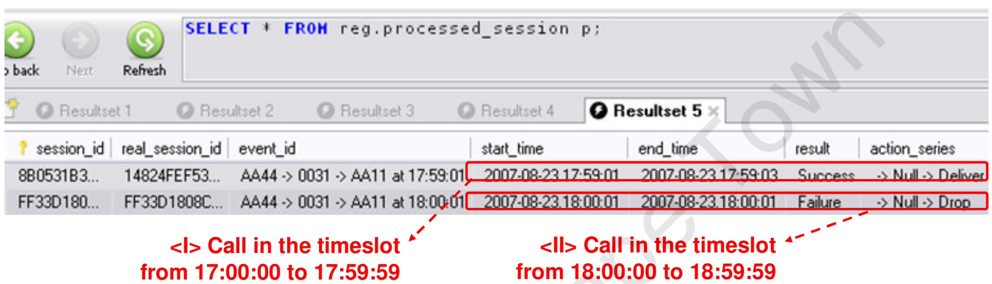


Figure 6-47. Java output to illustrate user schedule determining user availability.

Figure 6-47 exemplifies that the system makes different decisions at different processing moments for the same communication case, due to the change in the expected receiver's communication availability. When Deeya called Yang to her office phone at 17:59:01, Yang was still in her office and available on her "*phone*". The system would therefore immediately deliver the call to Yang (mark <II.i>). However, when Deeya repeated the call at 18:00:01, Yang's schedule indicated that she had moved from the "*office*" domain to the "*public*" domain (mark <I>). Because Yang only had "*GSM cell*" available in the "*public*" domain, the system had to fail Deeya's requested call (mark <II.ii>).

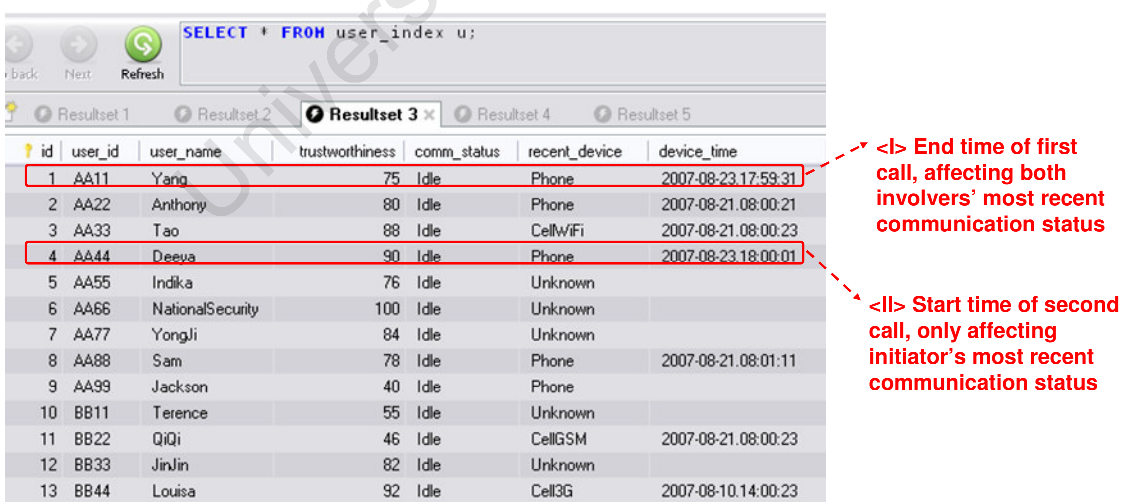


```
SELECT * FROM reg.processed_session p;
```

session_id	real_session_id	event_id	start_time	end_time	result	action_series
8B0531B3...	14824FEF53...	AA44 -> 0031 -> AA11 at 17:59:01	2007-08-23 17:59:01	2007-08-23 17:59:03	Success	-> Null -> Deliver
FF33D180...	FF33D1808C...	AA44 -> 0031 -> AA11 at 18:00:01	2007-08-23 18:00:01	2007-08-23 18:00:01	Failure	-> Null -> Drop

Figure 6-48. MySQL results about the effect of user schedule on communication availability.

Figure 6-48 further indicates that Deeya's call to Yang in the timeslot starting at 17:00:00 succeeded (mark <I>), whereas the call in the timeslot starting at 18:00:00 failed (mark <II>).



```
SELECT * FROM user_index u;
```

id	user_id	user_name	trustworthiness	comm_status	recent_device	device_time
1	AA11	Yang	75	Idle	Phone	2007-08-23 17:59:31
2	AA22	Anthony	80	Idle	Phone	2007-08-21 08:00:21
3	AA33	Tao	88	Idle	CellWiFi	2007-08-21 08:00:23
4	AA44	Deeya	90	Idle	Phone	2007-08-23 18:00:01
5	AA55	Indika	76	Idle	Unknown	
6	AA66	NationalSecurity	100	Idle	Unknown	
7	AA77	YongJi	84	Idle	Unknown	
8	AA88	Sam	78	Idle	Phone	2007-08-21 08:01:11
9	AA99	Jackson	40	Idle	Phone	
10	BB11	Terence	55	Idle	Unknown	
11	BB22	QiQi	46	Idle	CellGSM	2007-08-21 08:00:23
12	BB33	JinJin	82	Idle	Unknown	
13	BB44	Louisa	92	Idle	Cell3G	2007-08-10 14:00:23

Figure 6-49. Session-execution results for the cases in Table 6-12.

As shown in Figure 6-49, after Deeya attempted two calls to Yang at different moments, the latest times when the two users respectively used their devices showed different in the database. Deeya's first call at 17:59:01 succeeded and the system therefore updated the time when Deeya and Yang used their most recent devices to be the time when the first call ends – 17:59:31. Later, Deeya's second call at 18:00:01 failed. Because Deeya attempted to call Yang at 18:00:01, the most recent time when Deeya used her "*phone*" was updated at 18:00:01 (mark <II>). However, because Deeya's call failed to reach Yang, the most recent time when Yang used her "*phone*" remained at 17:59:31 (mark <I>).

6.4 Other Advantageous System Features and Several Limitations on Coding

Besides providing the above elementary functions, the system also enhances the network with several user-centric features such as the use of a graphic user interface (section 6.4.1) and the quantification of application performance (section 6.4.2).

6.4.1 Graphic User Interface Facilitating Human-Network Interaction

Graphic User Interfaces (GUI[49] [101]) facilitate the input and the modification of user communication profiles, including personal details (Figure 6-50), social relationships (Figure 6-51), and weekly schedules (Figure 6-52).

---

[49] GUI is the graphic user interface to a computer. A user controls a program via the use of icons, buttons, and pointers on the interface.

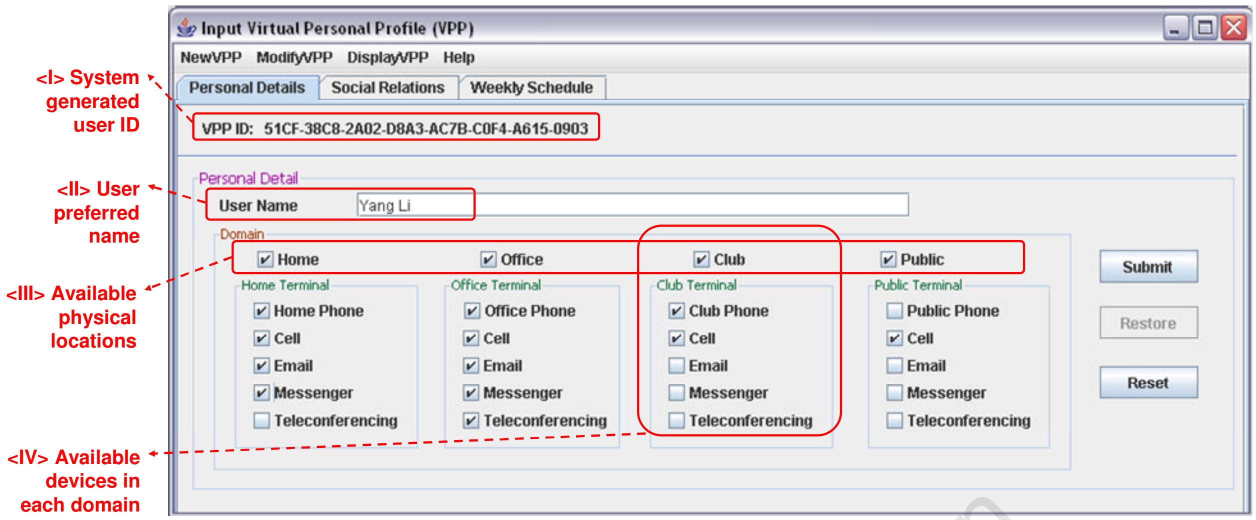


Figure 6-50. Virtual-Personal-Profile interface for inputting user personal details.

The current version of the virtual-user system has a simple Virtual-Personal-Profile GUI. Figure 6-50 shows the personal-detail input interface. The system automatically generates a unique user ID for a new user (mark <I>). Nevertheless, the user can type in his/her own preferred user name (mark <II>). The user can also choose the physical locations where most of his/her communications occur and the respective potential devices at those locations (marks <III> and <IV>). In future designs, the system is expected to provide a comprehensive list of physical locations and available devices so that the users can manually input the specific features of their devices, such as device number.

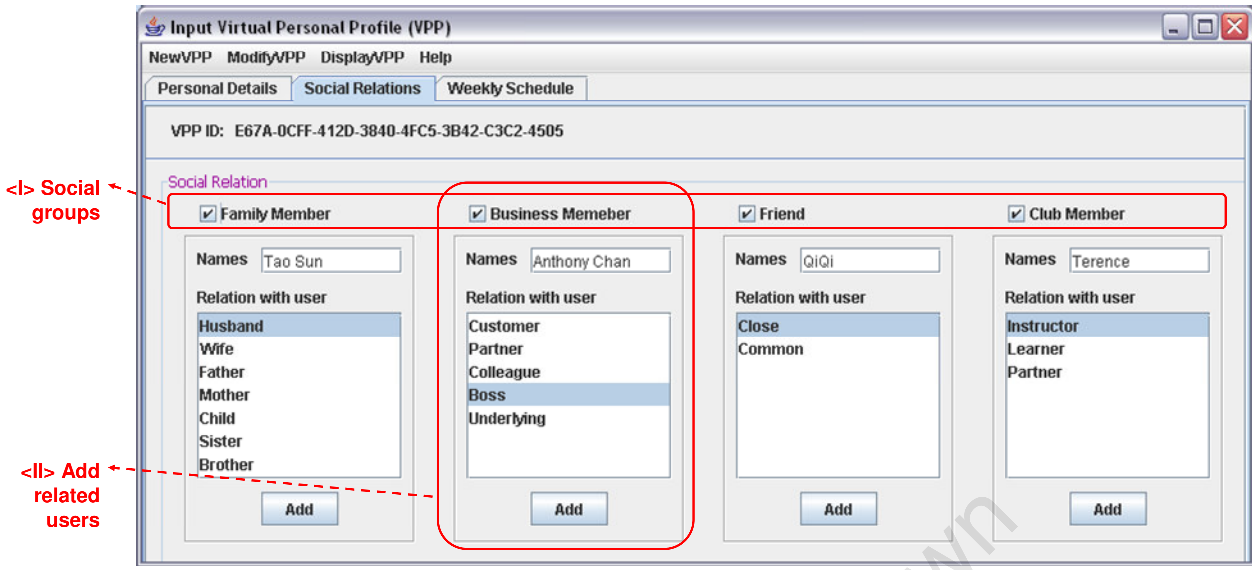
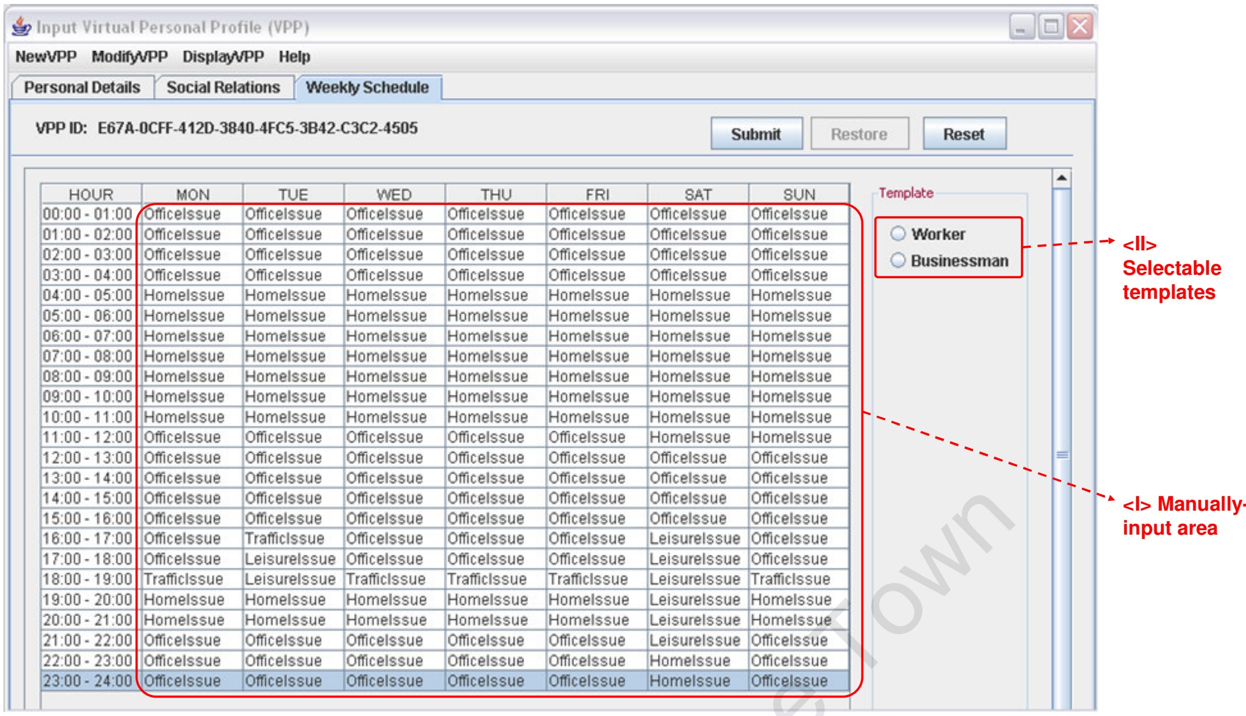


Figure 6-51. Virtual-Personal-Profile interface for inputting user social relations.

Figure 6-51 shows the social-relationship input interface. A user can add the related users from different social domains into its contact list through the interface (mark <I>). To add a related user, the main user first types the related user's name and then selects from the list below the most appropriate relationship type in which it relates to the added user (mark <II>). In future designs, the system is expected to provide a comprehensive list of social domains with plenty of relationship types in each domain. The main user should be able to select the related users from a user index with all communication users. In this way, the system gradually sets up a virtual-user network with all users relating to each other in social connections.



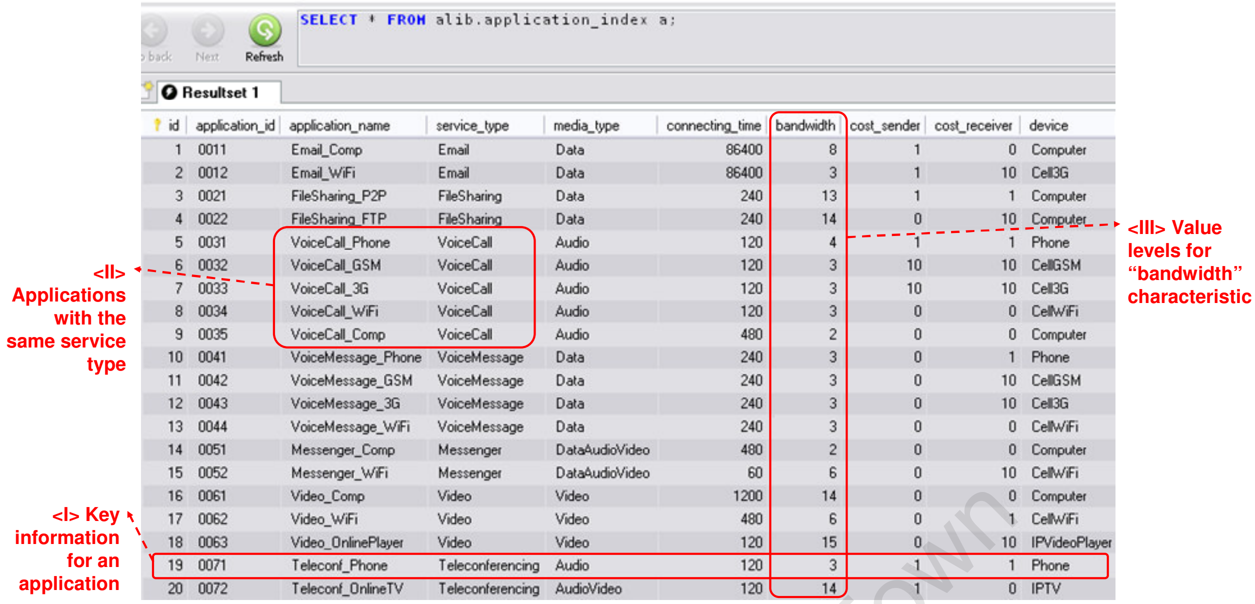
HOUR	MON	TUE	WED	THU	FRI	SAT	SUN
00:00 - 01:00	OfficeIssue	OfficeIssue	OfficeIssue	OfficeIssue	OfficeIssue	OfficeIssue	OfficeIssue
01:00 - 02:00	OfficeIssue	OfficeIssue	OfficeIssue	OfficeIssue	OfficeIssue	OfficeIssue	OfficeIssue
02:00 - 03:00	OfficeIssue	OfficeIssue	OfficeIssue	OfficeIssue	OfficeIssue	OfficeIssue	OfficeIssue
03:00 - 04:00	OfficeIssue	OfficeIssue	OfficeIssue	OfficeIssue	OfficeIssue	OfficeIssue	OfficeIssue
04:00 - 05:00	HomeIssue	HomeIssue	HomeIssue	HomeIssue	HomeIssue	HomeIssue	HomeIssue
05:00 - 06:00	HomeIssue	HomeIssue	HomeIssue	HomeIssue	HomeIssue	HomeIssue	HomeIssue
06:00 - 07:00	HomeIssue	HomeIssue	HomeIssue	HomeIssue	HomeIssue	HomeIssue	HomeIssue
07:00 - 08:00	HomeIssue	HomeIssue	HomeIssue	HomeIssue	HomeIssue	HomeIssue	HomeIssue
08:00 - 09:00	HomeIssue	HomeIssue	HomeIssue	HomeIssue	HomeIssue	HomeIssue	HomeIssue
09:00 - 10:00	HomeIssue	HomeIssue	HomeIssue	HomeIssue	HomeIssue	HomeIssue	HomeIssue
10:00 - 11:00	HomeIssue	HomeIssue	HomeIssue	HomeIssue	HomeIssue	HomeIssue	HomeIssue
11:00 - 12:00	OfficeIssue	OfficeIssue	OfficeIssue	OfficeIssue	OfficeIssue	OfficeIssue	HomeIssue
12:00 - 13:00	OfficeIssue	OfficeIssue	OfficeIssue	OfficeIssue	OfficeIssue	OfficeIssue	OfficeIssue
13:00 - 14:00	OfficeIssue	OfficeIssue	OfficeIssue	OfficeIssue	OfficeIssue	OfficeIssue	OfficeIssue
14:00 - 15:00	OfficeIssue	OfficeIssue	OfficeIssue	OfficeIssue	OfficeIssue	OfficeIssue	OfficeIssue
15:00 - 16:00	OfficeIssue	OfficeIssue	OfficeIssue	OfficeIssue	OfficeIssue	OfficeIssue	OfficeIssue
16:00 - 17:00	OfficeIssue	TrafficIssue	OfficeIssue	OfficeIssue	OfficeIssue	LeisureIssue	OfficeIssue
17:00 - 18:00	OfficeIssue	LeisureIssue	OfficeIssue	OfficeIssue	OfficeIssue	LeisureIssue	OfficeIssue
18:00 - 19:00	TrafficIssue	LeisureIssue	TrafficIssue	TrafficIssue	TrafficIssue	LeisureIssue	TrafficIssue
19:00 - 20:00	HomeIssue	HomeIssue	HomeIssue	HomeIssue	HomeIssue	LeisureIssue	HomeIssue
20:00 - 21:00	HomeIssue	HomeIssue	HomeIssue	HomeIssue	HomeIssue	LeisureIssue	HomeIssue
21:00 - 22:00	OfficeIssue	OfficeIssue	OfficeIssue	OfficeIssue	OfficeIssue	LeisureIssue	OfficeIssue
22:00 - 23:00	OfficeIssue	OfficeIssue	OfficeIssue	OfficeIssue	OfficeIssue	HomeIssue	OfficeIssue
23:00 - 24:00	OfficeIssue	OfficeIssue	OfficeIssue	OfficeIssue	OfficeIssue	HomeIssue	OfficeIssue

Figure 6-52. Virtual-Personal-Profile interface for inputting user weekly schedule.

Figure 6-52 shows the weekly-schedule input interface. A user can either manually input the most likely activity it performs in each hour during a week (mark <I>) or modifies from a template such as a “worker” template (mark <II>).

6.4.2 Reasonably Selected Characteristics for Application Performance

Another user-centric feature of the system is to present an application as several characteristics and quantify its performance as the level values of those characteristics. The quantified application performance makes it easier and faster for the service providers to identify an existing application or develop a new application because, to do so, the system only needs to compare the obtained real-world data with the level values stored in the database. Figure 6-53 gives an example of such a quantified application performance.



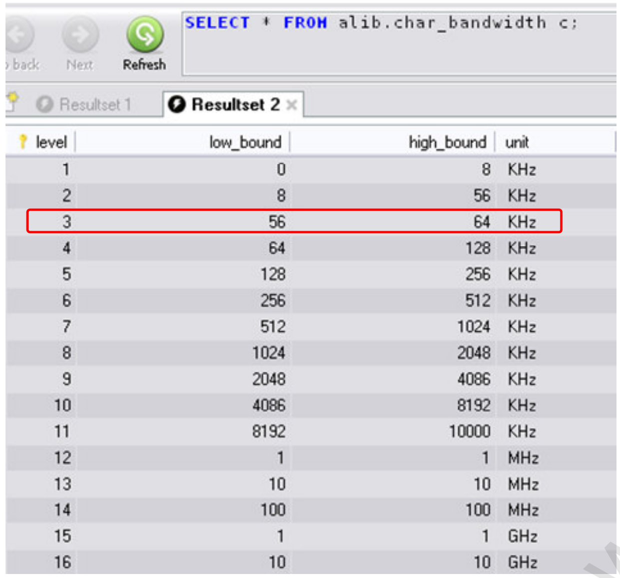
id	application_id	application_name	service_type	media_type	connecting_time	bandwidth	cost_sender	cost_receiver	device
1	0011	Email_Comp	Email	Data	86400	8	1	0	Computer
2	0012	Email_WiFi	Email	Data	86400	3	1	10	Cell3G
3	0021	FileSharing_P2P	FileSharing	Data	240	13	1	1	Computer
4	0022	FileSharing_FTP	FileSharing	Data	240	14	0	10	Computer
5	0031	VoiceCall_Phone	VoiceCall	Audio	120	4	1	1	Phone
6	0032	VoiceCall_GSM	VoiceCall	Audio	120	3	10	10	CellGSM
7	0033	VoiceCall_3G	VoiceCall	Audio	120	3	10	10	Cell3G
8	0034	VoiceCall_WiFi	VoiceCall	Audio	120	3	0	0	CellWiFi
9	0035	VoiceCall_Comp	VoiceCall	Audio	480	2	0	0	Computer
10	0041	VoiceMessage_Phone	VoiceMessage	Data	240	3	0	1	Phone
11	0042	VoiceMessage_GSM	VoiceMessage	Data	240	3	0	10	CellGSM
12	0043	VoiceMessage_3G	VoiceMessage	Data	240	3	0	10	Cell3G
13	0044	VoiceMessage_WiFi	VoiceMessage	Data	240	3	0	0	CellWiFi
14	0051	Messenger_Comp	Messenger	DataAudioVideo	480	2	0	0	Computer
15	0052	Messenger_WiFi	Messenger	DataAudioVideo	60	6	0	10	CellWiFi
16	0061	Video_Comp	Video	Video	1200	14	0	0	Computer
17	0062	Video_WiFi	Video	Video	480	6	0	1	CellWiFi
18	0063	Video_OnlinePlayer	Video	Video	120	15	0	10	IPVideoPlayer
19	0071	Teleconf_Phone	Teleconferencing	Audio	120	3	1	1	Phone
20	0072	Teleconf_OnlineTV	Teleconferencing	AudioVideo	120	14	1	0	IPTV

Figure 6-53. A fragment of a quantified application-performance index.

Figure 6-53 shows a simplified version of the application list used in the system. Each record contains the essential characteristics of an application, including application ID, application name, service type, media type, connecting time, required bandwidth, cost to initiator, cost to receiver, and potential device (mark <I>). The value type of these characteristics can be string (i.e., for application name), enumeration (i.e., for media type), integer (i.e., for connecting time), and index (i.e., for bandwidth).

What is worth mentioning is that, when Session Generator looks for potentially available applications for a session, it collects the applications that share the same service type as the expected one (mark <II>).

When the description of a characteristic requires more than one type of value, the system first assigns an index value to the characteristic and then describes the characteristic in detail in other tables. For example, the "*bandwidth*" characteristic indicated by an integer index (mark <III> in Figure 6-53) needs three values to describe each index level in detail, as shown in Figure 6-54.


back Next Refresh `SELECT * FROM alib.char_bandwidth c;`

Resultset 1 Resultset 2

level	low_bound	high_bound	unit
1	0	8	KHz
2	8	56	KHz
3	56	64	KHz
4	64	128	KHz
5	128	256	KHz
6	256	512	KHz
7	512	1024	KHz
8	1024	2048	KHz
9	2048	4086	KHz
10	4086	8192	KHz
11	8192	10000	KHz
12	1	1	MHz
13	10	10	MHz
14	100	100	MHz
15	1	1	GHz
16	10	10	GHz

Figure 6-54. Detailed description of the quantified "bandwidth" characteristic.

Figure 6-54 describes a "*bandwidth*" value in three sub-values of low bound, high bound, and unit. For example, if the level value of an application's bandwidth in Figure 6-53 is "3", then the application requires a bandwidth with the range from 56 kHz to 64 kHz.

6.5 Scope of Programming Social Networking

Although the current software realization of the virtual-user system is able to conduct part of the human-like intelligence we expect, it has several intrinsic limitations in implementing full human-like intelligence and real-world social networking and needs an improvement.

Firstly, software coding lacks the capability of matching social dynamics and the flexibility in presenting social diversity. To operate social networking is to conduct a unpredictable information-exchange process. Even the most detailed software design can hardly cover all social-networking scenarios, to say nothing of the factual software coding. In addition, it requires great effort to maintain and upgrade software codes so that they function in accordance with the timely changes in human society. Hereof, explicit data structures in design and sufficient network resources in reality are necessary for a software environment to handle the complexity of a social system.

Secondly, the simulation scenarios are only based on a small-scaled personal network – Yang's social network and the experimental results therefore have limitation. Due to the small size and simple topology of the network example, the experimental results cannot be directly applied to large-scaled telecommunications network that manages complex user relations. Only if we use real-world data as input in an emulation environment, would we be able to obtain practical experimental data and optimize the system accordingly.

Thirdly, the selected categories of analysis cannot cover all expected human-like intelligence. We have only validated the functionality of the system modules and their abilities of determining optimal session-delivery manner. The validation of several other essential intelligence abilities such as system self-organization are still yet to accomplish.

Lastly, the above validation has just proved the realizability of the intelligence approach in a software environment. Many more questions need an answer if we put the system into industrial practice. Is the system realizable in hardware? Can the system process a large amount of real-world data? What more features does the system needs to import? Is the system able to provide better services compared with what the existing network intelligence has achieved?

6.6 Summary of Chapter 6

The output Java results and the corresponding MySQL records on session-delivery decision-making in this chapter have successfully validated the functionality of the virtual-user system. Firstly, the stepped session decision-making results have proven that each function module of the system was able to function independently and fulfil the tasks. Session Registrar properly abstracted and stored key session information. Session Generator created virtual-session pairs by correctly understanding the requirements and then fetching the needed information from relevant modules. Using these virtual-session pairs, Session Comparator generated valuable comparison results for any two virtual sessions in a pair. Decision Maker was able to make decisions about optimal session delivery according to a real-time communication environment. Session Keeper stored the sessions if needed and invoked other modules to reprocess the sessions at the right moments. All these function modules functioned collaboratively to implement the manners of *fail*, *deliver*, *force*, *postpone*, *help*, and *learn* for various communication scenarios by effectively using the available communication resources. We further proved that the users' social features positively influenced the intelligence decision-making on session delivery through the quantification of their social relationship and the collaboration of their trust degree with trustworthiness. Finally, the user-friendly GUI and the quantified application performance enhanced the system with user-centric properties.

Chapter 7 will address the improvement in communication efficiency that the system can bring to the communications network. The focus will be on session-delivery success rate, network traffic, network resource utilization, new network abilities, and time delay in decision-making.

Chapter 7 Improving Communication Efficiency via the Virtual-User System

Continuous developments in technology have endowed current communications network with the capability to provide outstanding services. These services guarantee adequate bandwidth, high-speed transmission rate, negligible delays, assured security, and reasonable cost charge to users. All these features are very enjoyable when the users are accessible to the services. However, if the users are personally not available to the services, they cannot enjoy any of these features no matter how well the services can perform.

The proposed virtual-user system is solving the user-unavailability problem at the human side. It increases the users' availability to communication services by making the most of existing network- and human-resources as well as involving new human resources via the users' social relationships. The system assists the communications network to improve the success rate of communication sessions (section 7.2), smoothen network traffic (section 7.3), sufficiently utilize valid network- and human-resources (section 7.4), and acquire the intelligent abilities of self-learning and dynamic storage (section 7.5). These benefits convey the intelligence to the network. Gaining them only costs a small overhead of increased time delay during the decision-making for optimal session delivery (section 7.6).

7.1 Preconditions for System Evaluation

To give prominence to the benefits of users' social issues to the network, we make the following assumptions before evaluating the virtual-user system. (1) Under normal circumstances, only two users in a one-degree connection (section 2.1.2) can directly contact each other. That is, a user can only contact another user when the former is in the preset contact list of the latter. (2) Under intelligence circumstances, two users in a two-degree connection are also able to contact each other but via their commonly related users, using the intelligence action of *help* or *learn*. For simplicity purposes, we restrict the contactable users to within a two-degree connection in this chapter. (3) Under intelligence circumstances, the users in a one-degree connection can also use two other intelligence actions – *force* and *postpone* – according to their communication status and preference. (4) We select Yang's social network (Figure 6-1) as the sample network for a system evaluation with all the sessions below happening between any two users in the network.

We call a successful session delivery under normal circumstances a "general delivery" and under intelligence circumstances an "intelligence delivery". The intelligence delivery thus includes general delivery, delivery by forcing, delivery by postponing, delivery by help, and delivery by learning, which are respectively shortened according to the delivery manner as *deliver*, *force*, *postpone*, *help*, and *learn* in what follows. We express the failure of delivering a session as the manner of *fail*. Correspondingly, we call the network with intelligence embedded an "intelligence network" and the network without intelligence embedded a "general network".

To testify to system performance, it is necessary to first generate different types of network traffic, represented by "session arrival rate". We define the session arrival rate as the number of session-service requests per unit time. This rate is time dependent and the sessions should accommodate those that have been or need to be processed. In what follows, we use the number of session-service requests generated at a discrete moment to represent the session arrival rate over a unit period starting from that moment. These discrete moments are distributed evenly in time (i.e., one moment every 15 seconds) and the number of requests at those moments abides by a probability distribution. Analogous to the definition of the session arrival rate, we define "session delivery rate" as the number of actually delivered sessions per unit time, where these sessions have initially been requested at the starting moment of that unit time. For simplicity, we assume that the execution duration of all sessions is a constant (i.e., 30 seconds). All these parameters can be adjusted to suit real world communication traffic.

It is understood that we built the above preconditions only for the prototype on academic evaluation (i.e., for proof-of-concept), expecting the experimental results to be able to properly illustrate system behaviors (not to provide the sources for accurate measurement on system performance). Therefore, we analyze single-run experimental data instead of averaged multi-run data because the former is able to explicitly exhibit system behaviours whereas the latter complicates the exhibition. There is still a long way to go to promote this academic prototype to industrial practice in terms of performance.

7.2 Improved Success Rate of Communication Sessions

Delivery success rate of communication sessions is the number of successfully delivered sessions in a period divided by the number of expected-to-be-delivered sessions (i.e., session arrival rate) in that period. Under intelligence circumstances, the calculation of the success rate considers the successful deliveries using both general and intelligence manners.

7.2.1 Success Rate over Short Term

We first generate short-term network traffic, with its session arrival rate being determined by the information provided in Table 7-1. We then present the traffic and the results obtained from traffic processing in Figure 7-1 to show whether an intelligence network is able to perform better than a general network.

Table 7-1. Communication scenario for session-delivery success rate over short term.

Variable	Unit	Content	Notes
distribution type	N/A	Poisson distribution[50] ([102])	In amplitude, x_0=0, lambda=5
traffic type	N/A	Real-time traffic	E.g., voice call
session-execution start time	SystemTime	2007-08-10.14:00:01	
number of units	N/A	20	
interval of each unit	Second	15	
sequential number of peak unit	N/A	8th	
expected number of sessions in peak unit	N/A	5	lambda=5
duration of session execution	Second	30	

* x_0: Start value of the number of occurrences.

* lambda: Expected number of occurrences during the given interval.

* N/A: Short for “not applicable”.

Table 7-1 provides the information to construct a communication scenario running over five minutes, with 20 time units of 15 seconds each. The system generates session-service

[50] Poisson distribution is a discrete probability distribution that expresses the probability of a number of events occurring in a fixed period if the event arrival rate is fixed and the event occurrence is time independent.

requests at the beginning of each unit. The probability of generating a fixed number of requests at a time is Poisson distributed, with an average of five session-service requests made in a peak-time unit. By assigning the time as the dependent variable and the number of session-service requests at a moment as the independent variable, the system generates a type of network traffic as shown in Figure 7-1:

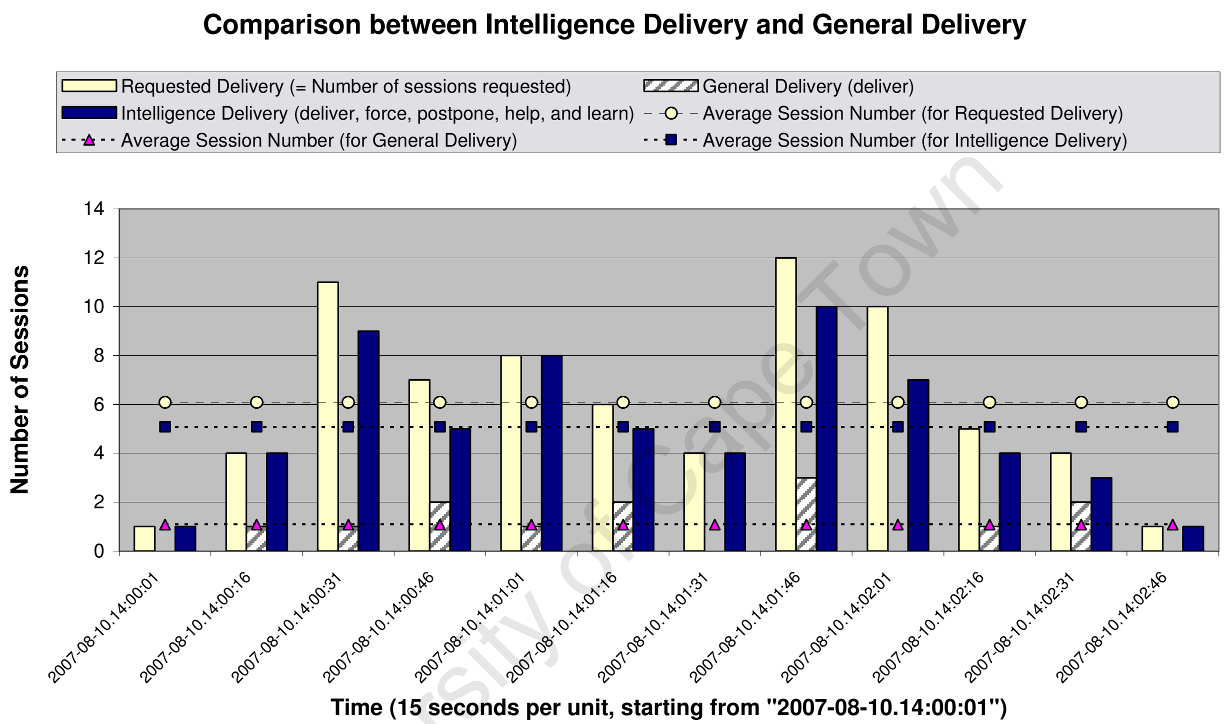


Figure 7-1. Comparing the intelligence delivery with the general delivery.

Figure 7-1 exhibits how successful a network is able to deliver the service sessions. In this experiment, the input to the network at successive moments is shown as the number of sessions requested to be delivered (light-colour bars) with an average value of "six" (dot-marked line). This figure then compares the number of sessions actually being delivered without intelligence involved (bars with strips) with that actually being delivered with intelligence involved (dark-colour bars) at each moment. The average session number obtained from the general delivery (without intelligence involved) is only "one" shown in triangle-marked line, which can hardly meet the expectation of "six". In contrast, the average session number obtained from the intelligence delivery is “five” shown in square-marked line, which is close to the ideal number.

Our experiment shows that an intelligence network is able to process most session requests and surpasses the general networks. Figure 7-1 has demonstrated this result in terms of the successful delivery of sessions. Figure 7-2 further demonstrates this result in terms of the success rate of session delivery.

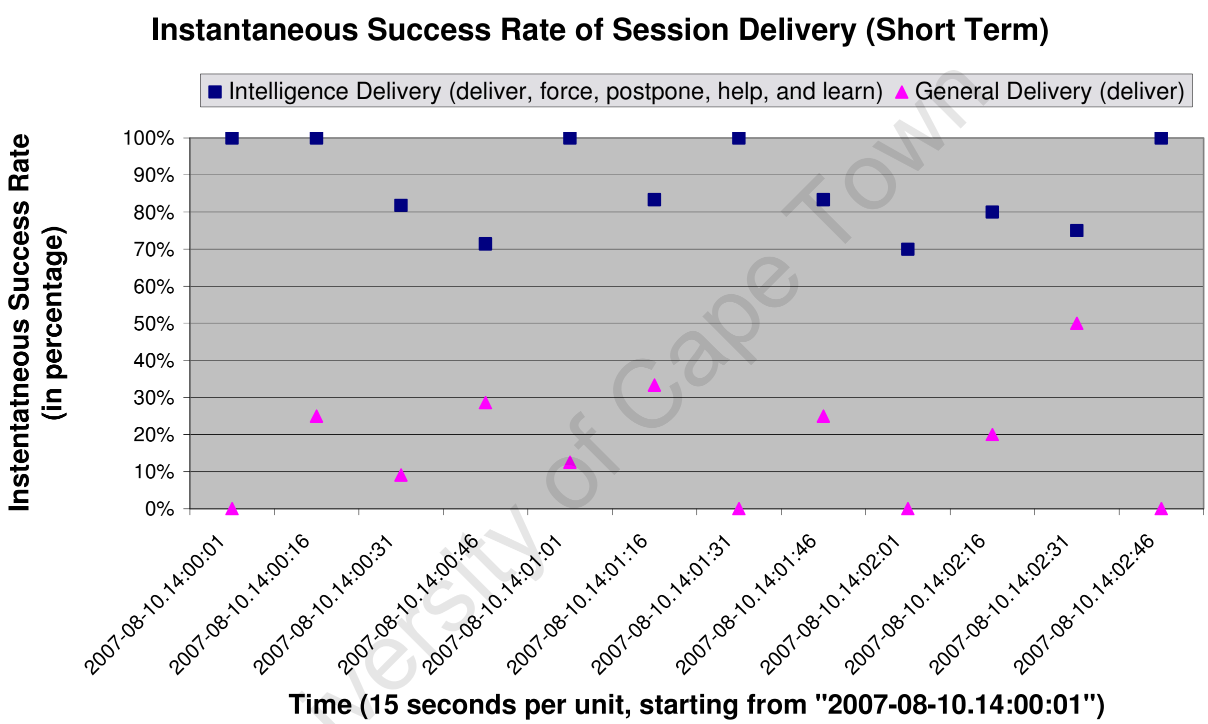


Figure 7-2. Increased success rate of session delivery over short term.

Figure 7-2 compares the success rate of session delivery under the intelligence scenario (marked by squares) with that under the normal scenario (marked by triangles). Although the intelligence network cannot deliver all the sessions as requested, it provides a satisfying average success rate of 87.08%, which is four times that of the general network at 16.96%. The success of the intelligence network can be attributed to the fact that it performs four extra actions that *force*, *postpone*, *help*, and *learn* to save the would-be-failed sessions.

In the real world, the delivery success rate of a general network is much higher than that shown in Figure 7-2. We have previously made some specific assumptions when generating the session arrival rate of the general network (section 7.1) to explicitly exhibit the improvement that the intelligence network makes upon the general one. (1) One preset rule regulates that a session is to fail when the expected session initiator and receiver do not have a direct connection. This assumption is beneficial by filtering unnecessary aggressive sessions for the receiver and therefore safeguarding his/her personal network. Nevertheless, it brings inconvenience to the receiver by limiting his/her opportunities of being reached. Another intelligence ability called self-learning (section 7.5.1) is able to make up for this shortcoming. Through the learning process, the network is able to set up sessions between indirectly connected users through their common related user and individual preferences. (2) The results in Figure 7-2 are based on Yang's social-relationship topology that is star shaped with Yang as the centre (Figure 6-1). Only a few other users are directly connected. Because Yang plays the role of setting up most connections for indirectly connected users, the probability of her communication status being busy is very high. This causes a high failure rate of session delivery when rule (1) applies. However, other topologies such as net would have a better session-delivery success rate for the general network compared to Figure 7-2 because most users are directly connected.

7.2.2 Success Rate over Longer Term

Under the communication scenario described in Table 7-2, Figure 7-3 discloses the extent to which an intelligence network overcomes a general network in terms of the instantaneous success rate of session delivery over the longer term.

Table 7-2. Communication scenario for session-delivery success rate over longer term.

Variable	Unit	Content	Notes
distribution type	N/A	Poisson distribution	In probability, x_0=0; lambda=5
traffic type	N/A	Real-time traffic	E.g., voice call
session-execution start time	SystemTime	2007-08-10.14:00:01	
number of units	N/A	240	1 hour
interval of each unit	Second	15	

average number of sessions per unit	N/A	5	lambda=5
duration of event execution	Second	30	

Table 7-2 provides the information to construct a communication scenario running over one hour, with 240 time units of 15 seconds each. The system generates session-service requests at the beginning of each unit. The probability of generating a fixed number of requests at a time abides by the Poisson distribution, with an average of five session-service requests made in each unit. The scenario described in Table 7-2 assigns the time as a dependent variable and the number of session-service requests expected at a specific moment, generated in a Poisson distribution, as an independent variable. We get the instantaneous success rates of the general and intelligence deliveries at successive moments over a longer term in Figure 7-3:

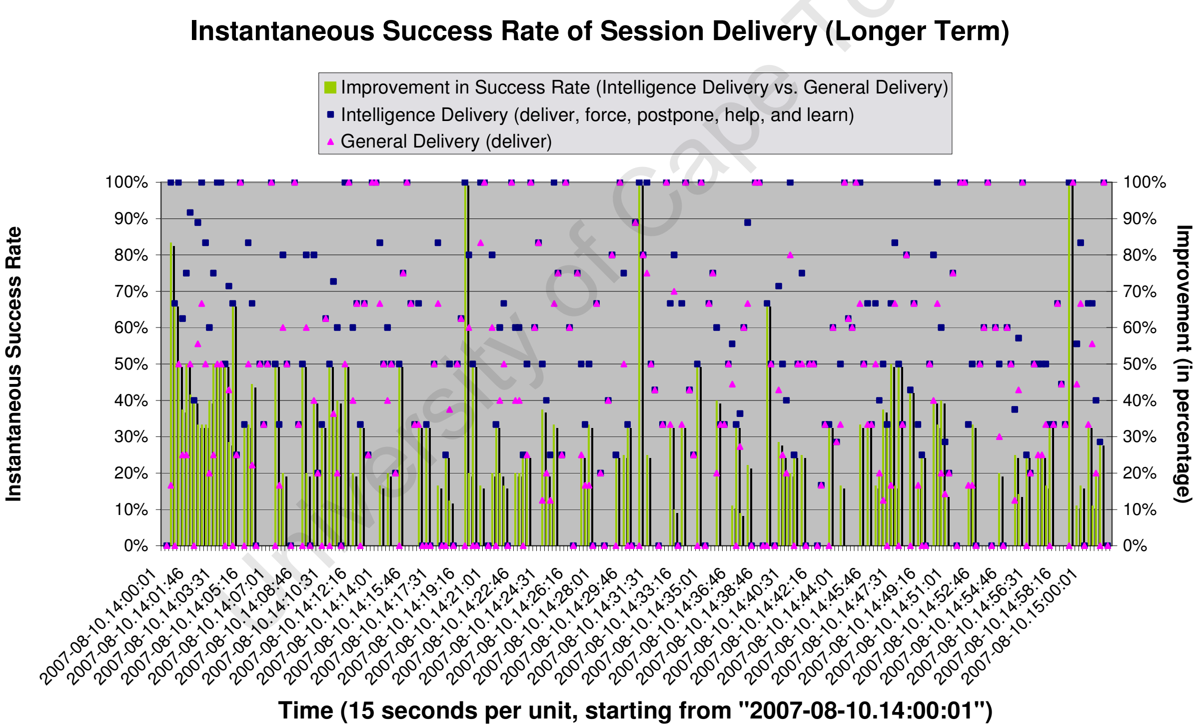


Figure 7-3. Increased success rate of session delivery over the longer term.

The title "instantaneous success rate" by the y-axis on the left and the corresponding axis values illustrate the success rate for both Intelligence Delivery (marked by squares) and General Delivery (marked by triangles). The title "improvement (in percentage)" by the y-axis on the right and the corresponding axis values illustrate of the achievement that Intelligence Delivery made in success rate when compared with General Delivery (marked by bars).

Figure 7-3 compares the success rate of the intelligence delivery (marked by squares) with that of the general delivery (marked by triangles) at successive moments over a longer term. The improvement in the instantaneous success rate from the general delivery to the intelligence delivery is shown in shadowed bars. This figure discloses two issues. Firstly, the virtual-user system possesses consistency in providing intelligence delivery over a longer term. In this experiment, the system is able to successfully deliver a significant number of likely-to-fail-in-general-scenario sessions through the intelligence delivery about every 60 seconds. Secondly, the system possesses continuity in terms of improving session-delivery success rate through the intelligence delivery. In the experiment, an average improvement in the instantaneous success rate from the general delivery to the intelligence delivery is 15.56% of the total, which nearly equals half of the general delivery whose success rate averages at 38.34%.

To sum up the sections 7.2.1 and 7.2.2, the virtual-user system helps improve network performance by increasing the session-delivery success rate of a general delivery by around 70.12% at the initial stage and around 15.56% in the longer term.

7.3 Well-balanced Network Traffic

Narrowly speaking, network traffic is a one-dimensional matrix[51] ([103]) with each element being the number of sessions that the network requests or delivers at a time. The requested traffic and the actually generated traffic are respectively in accordance with the session arrival rate and the session delivery rate. Despite the difference between various traffic types, two generally accepted principles are that, at any moment, (1) the more sessions the network is handling, the heavier the traffic is and (2) the less fluctuating the traffic is, the more stable the network is. In this section, the actual deliveries refer to the actual intelligence deliveries.

7.3.1 Balanced Real-time Traffic

[51] Matrix is an array of elements, generally used for linear transformation in mathematics and statistics.

Under the communication scenario described in Table 7-3, Figure 7-4 shows the requested and actual deliveries of sessions that carry real-time traffic over time.

Table 7-3. Communication scenario for real-time network traffic.

Variable	Unit	Content	Notes
distribution type	N/A	Poisson distribution	In amplitude , x_0=0, lambda=5
traffic type	N/A	Real-time traffic	E.g., voice call
session-execution start time	SystemTime	2007-08-10.14:00:01	
number of units	N/A	8	
interval of each unit	Second	15	
sequential number of peak unit	N/A	4th	
expected number of sessions in peak unit	N/A	5	lambda=5
duration of session execution	Second	30	

* x_0, lambda, and N/A: Same as the above "x_0", "lambda", and "N/A".

Table 7-3 provides the information to construct a communication scenario where the network is carrying real-time voice-call traffic. The requested traffic and the actually generated traffic are compared in Figure 7-4:

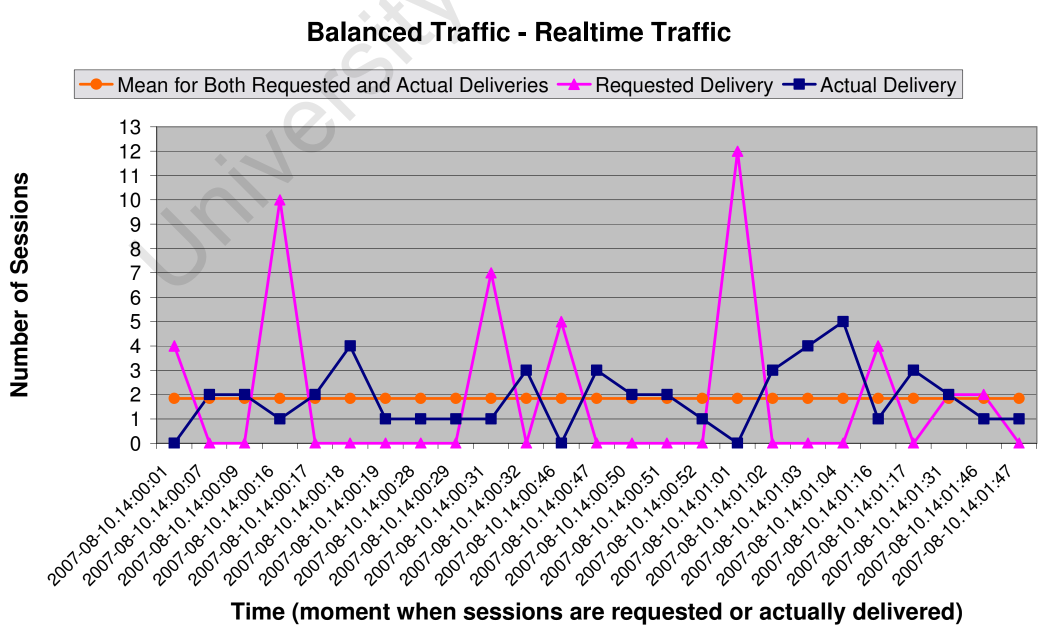


Figure 7-4. Balanced real-time network traffic through intelligence delivery.

The time is scaled unevenly in value but evenly according to the moments when the sessions occur.

For the real-time voice-call traffic generated according to Table 7-3, we compare its requested shape (marked by triangles) with its actual-delivery shape (marked by squares) in Figure 7-4 and get two positive conclusions. Firstly, the actual session delivery is smoother than the theoretically requested one. The smoothened traffic reduces the number of traffic pulses and propounds an easier control of traffic for network operators. Secondly, after the network has adopted intelligence, the maximum number of sessions that need executing at a time decreases from 12 to 5. This makes the network-bottleneck problem less harsh for hardware designers and manufacturers.

Figure 7-5 shows more clearly that the actually delivered traffic is smoother than the requested one by presenting the deviation of session numbers from mean at the discrete moments when the sessions occur (i.e., being requested or being actually delivered):

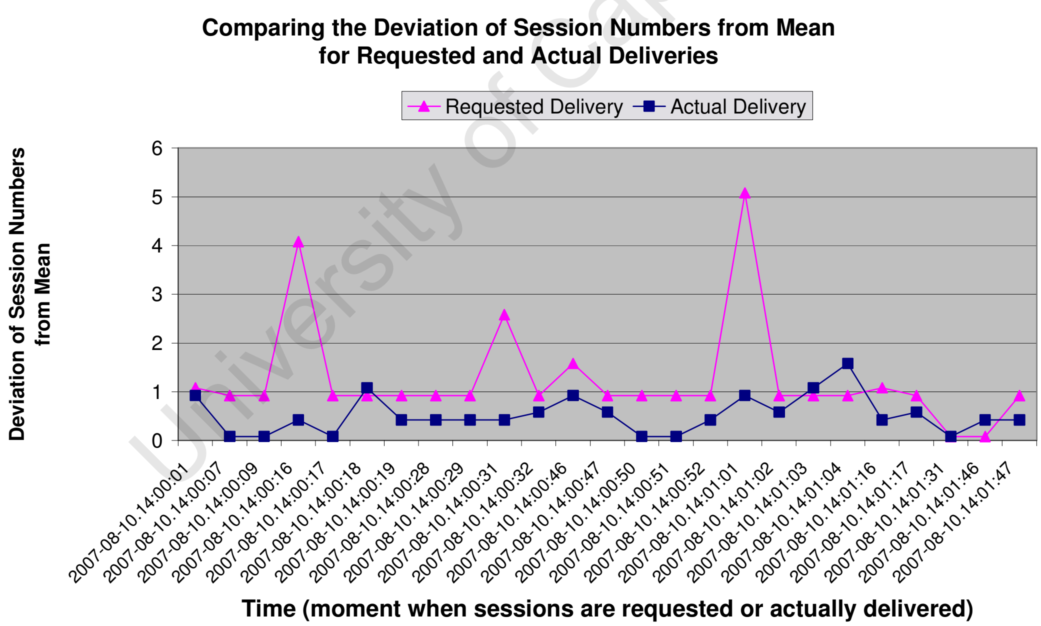


Figure 7-5. Deviation of session numbers from mean for requested and actual deliveries.

The average number of sessions at each moment is 1.84 in Figure 7-4 for both the requested and actual deliveries. Based on this, Figure 7-5 shows that, at most moments, the requested delivery deviates further from the average value – 1.84 – than the actual delivery does.

The deviation of the requested delivery averages at 1.2512 while that of the actual delivery is only 0.5232. The system has therefore reduced the fluctuation of the session delivery to 41.8%.

7.3.2 Balanced General Traffic

Under the communication scenario described in Table 7-4, Figure 7-6 shows the requested and actual deliveries of sessions that carry general traffic over time.

Table 7-4. Communication scenario for general network traffic.

Variable	Unit	Content	Notes
distribution type	N/A	Poisson distribution	In amplitude, x_0=0, lambda=5
traffic type	N/A	General traffic	E.g., Email, voice call, file sharing, teleconferencing, etc.
session-execution start time	SystemTime	2007-08-10.14:00:01	
number of units	N/A	8	
interval of each unit	Second	15	
sequential number of peak unit	N/A	4^{th}	
expected number of sessions in peak unit	N/A	5	lambda=5
duration of event execution	Second	30	

Table 7-4 provides the information to construct a communication scenario where the network is running general network traffic that contains different media types, including data, audio, video, mixed data and audio, mixed audio and video, and mixed data, audio, and video. The general traffic thus has variable requirements in connection rate, data flow, transmission delay, etc. on the sessions with different media types. Figure 7-6 compares the requested shape of such traffic with its actual-delivery shape:

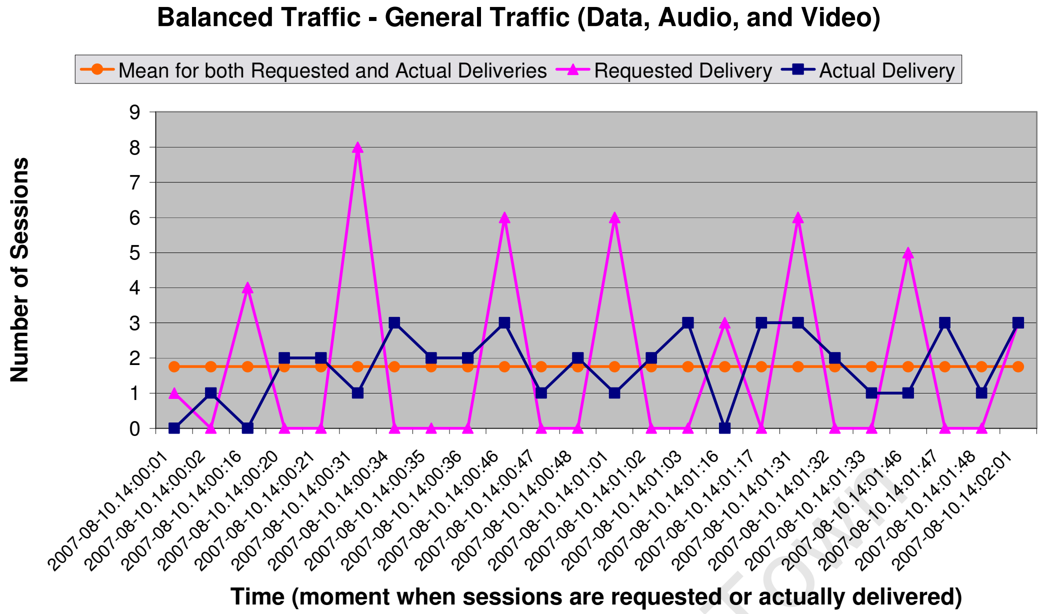


Figure 7-6. Balanced general network traffic through intelligence delivery.

Compared with the requested real-time traffic in Figure 7-4, the requested general traffic in Figure 7-6 is more stable by itself with the inclusion of different media types. Even so, the intelligence network is still able to constrain the general traffic to a smoother shape, resulting in more even traffic with less harsh bottlenecks. The average number of sessions occurring at each moment is 1.75 (viz., mean), the deviation of the requested delivery from the mean is 1.13, and that of the actual delivery is 0.44. The system has therefore reduced traffic fluctuation to 38.94%.

To sum up the sections 7.3.1 and 7.3.2, the virtual-user system helps improve network performance by balancing network traffic, regardless of the type of media it carries, by diminishing the fluctuation of the traffic to around 40% of the original one in terms of deviation.

7.4 Optimum Utilization of Available Communication Resources

In theory (Chapter 3), the intelligence network is able to make optimum use of network resources via intelligence actions. These usages include utilizing extended immediately available network resources via *force*, making the most of expected human resources via *postpone*, making use of spare human resources via *help*, and exchanging information resources between humans and the network via *postpone*, *force*, *help*, and *learn*. Thus if the virtual-user system succeeds in executing sessions through any of these actions, the intelligence network is making greater use of available network resources than a general one.

Under the communication scenario described in Table 7-5, the series from Figure 7-7 to Figure 7-11 illustrate the cumulative number of successfully processed sessions via all manners of *deliver*, *force*, *postpone*, *help*, *learn*, and *fail.*

Table 7-5. Communication scenario for optimum utilization of available network resources.

Variable	Unit	Content	Notes
distribution type	N/A	Poisson distribution	In probability, x_0=0, lambda=5
traffic type	N/A	Real-time traffic	E.g., voice call
session-execution start time	SystemTime	2007-08-10.14:00:01	
number of units	N/A	240	1 hour
interval of each unit	Second	15	
average number of events per unit	N/A	5	lambda=5
duration of event execution	Second	20	

* x_0, lambda, and N/A: Same as the above "x_0", "lambda", and "N/A".

Table 7-5 provides the information to construct a communication scenario running over one hour, with 240 time units of 15 seconds each. The system generates traffic at the beginning of each unit. The probability of generating a fixed number of session-service requests at a time abides by the Poisson distribution, with an average of five requests expected in each unit. Table 7-5 assigns the time as the dependent variable and the number of session-service requests made at a given moment as the independent variable.

7.4.1 Micro Features of the Network-Resource Utilization

Figure 7-7 and Figure 7-8 illustrate the cumulative number of sessions that are processed using different actions. Figure 7-7 addresses the initial stage when the intelligence network starts to process communication sessions and Figure 7-8 intercepts an in-process stage when the network has been operating for around one hour.

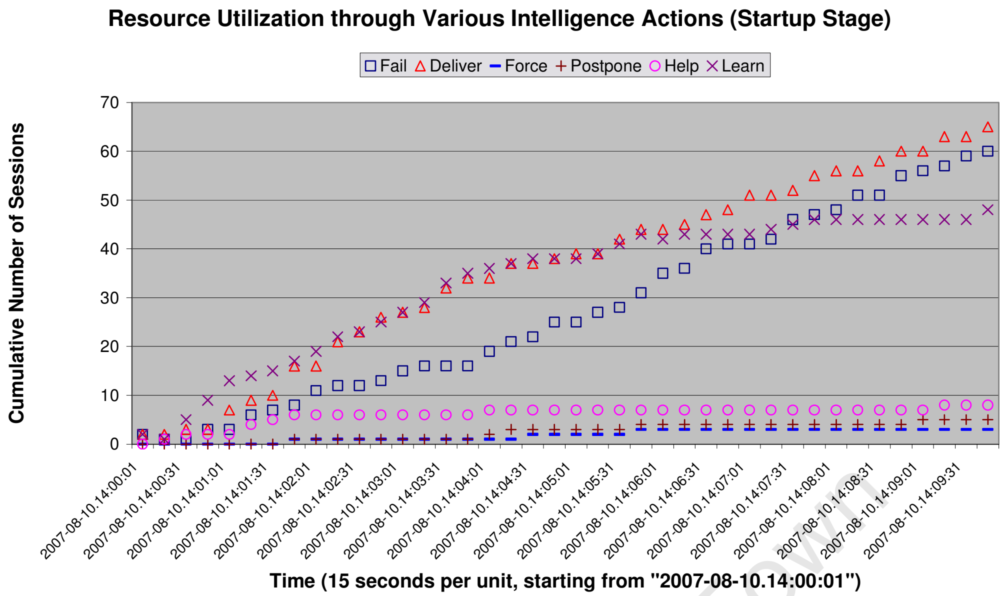


Figure 7-7. Resource utilization through general and intelligence actions at start-up stage.

Figure 7-7 shows how the intelligence network effectively distributes network resources to facilitate session delivery in the first ten minutes of the observation.

The cumulative numbers of sessions delivered by the manners of *deliver* (marked by triangles) and *fail* (marked by rectangles) increase linearly with time, showing that these intelligence manners have stable effect on session delivery.

The manners of *force* (marked by dashes), *postpone* (marked by pluses), and *help* (marked by circles) contribute little to total session delivery at the start-up stage. That is, the network has only used a limited amount of the extended network resources and spare human resources. Among the successful deliveries by these manners, the majority comes from the manner of *help*. The observation indicates that the network relies more on user social relationships than on extended network resources at the beginning for the successful delivery of communication sessions.

The manner of *learn* assists in delivering a large percentage of sessions at every moment in the first 10 minutes, owing to its capability to provide the network with more attempts at session delivery. The large number of *learn* actions also implies that a network is normally busy with learning users' social relationships and preferences at the start-up stage. These actions result in the exchange of a massive amount of information resources between the social network and the communications network.

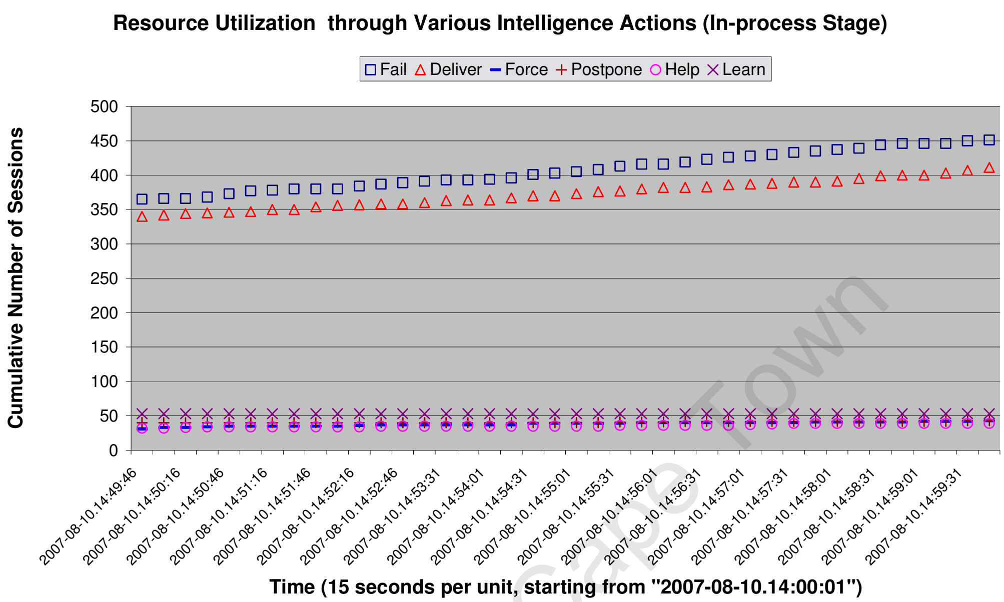


Figure 7-8. Resource utilization through general and intelligence actions at in-process stage.

Figure 7-8 shows how the intelligence network distributes communication resources to assist session delivery in the last 10 minutes of the observing hour. With time, the behaviour of all actions gradually becomes stable. The majority of sessions are delivered through the two non-intelligence actions – *deliver* and *fail,* which is in accordance with the session-delivery situation of the communications network. Other than that, the intelligence network also executes the intelligence actions of *force*, *postpone*, *help*, and *learn* to save those would-be-failed-under-normal-circumstance sessions. The numbers of these processed-through-intelligence sessions are quite small when compared with those of generally processed sessions and they increase slowly.

7.4.2 Macro Features of the Network-Resource Utilization

In observing the session delivery for one hour as a whole, we can identify several macro features of resource utilization. Figure 7-9 exhibits the cumulative number of sessions delivered through different actions over time with both the number and the time according to previous scales. Figure 7-10 does so with the number on a logarithmic scale, whereas Figure 7-11 depicts the numbers using stacked lines. For all these figures, we exclude the session failure (by the manner of *fail)* but include the intelligence delivery that sums up the number of sessions delivered through all actions of *deliver*, *force*, *postpone*, *help*, and *learn*.

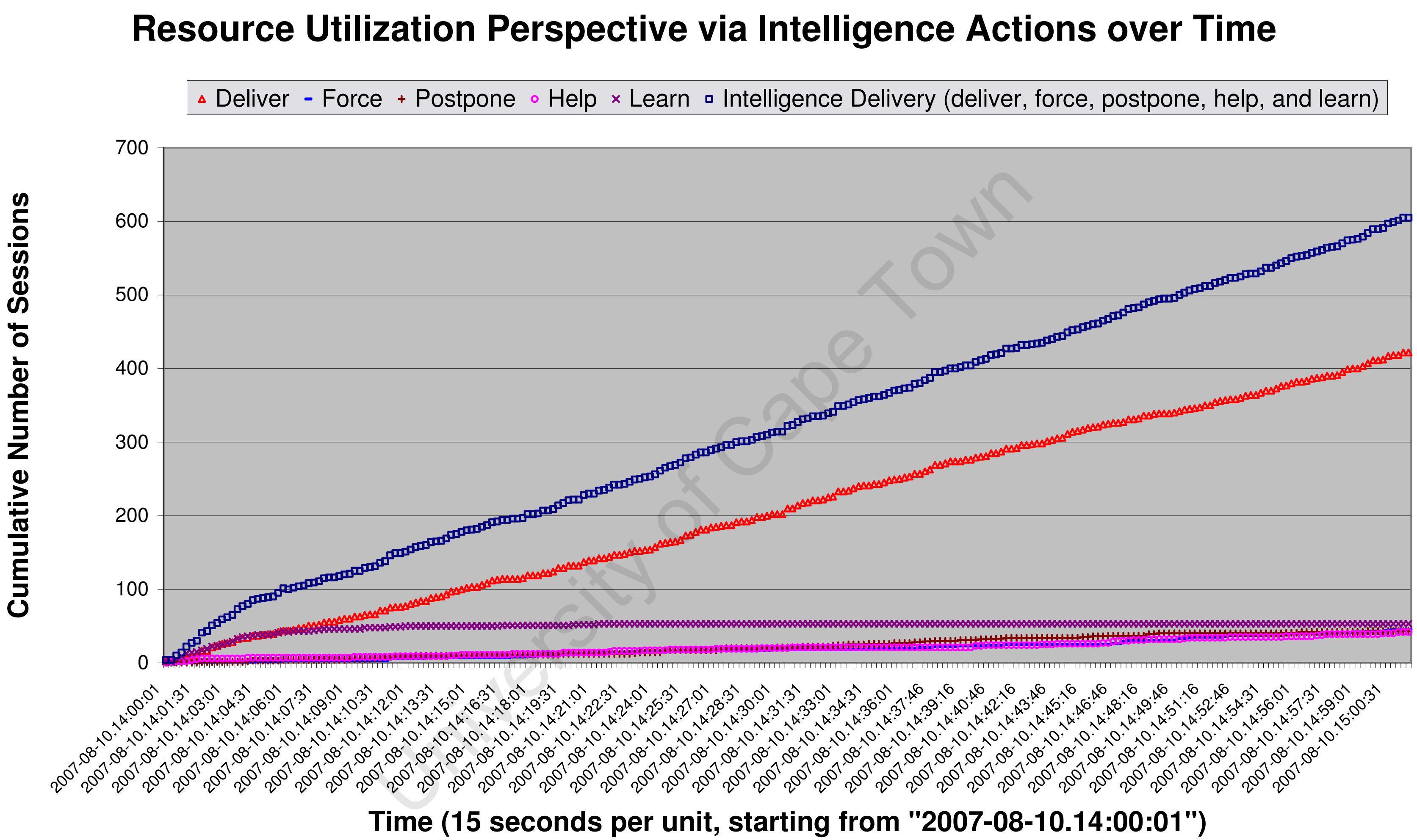


Figure 7-9. Resource utilization through general/intelligence actions over longer term (1).

Y-axis is in linear scale. Phenomena hidden in the intersected marks at the bottom will be further explained in Figure 7-10 and Figure 7-11.

Figure 7-9 shows the cumulative numbers of successfully delivered sessions through different actions for one hour. The number of sessions delivered through intelligence delivery (marked by rectangles) at a given moment is the sum of the numbers obtained through the actions of *deliver* (marked by triangles), *force* (marked by short lines), *postpone* (marked by crossings), *help* (marked by circles), and *learn* (marked by diamond-crossings) at that moment.

The number of successfully delivered sessions increases with time, regardless of whether the delivery types of these sessions are intelligence delivery or general delivery. The intelligence delivery successfully delivers more sessions than the general delivery at all time. Furthermore, the number of sessions through intelligence delivery accelerates faster than through general delivery due to the increasing combined contribution from the intelligence actions. In other words, the absolute number of sessions saved by intelligence actions (deducting the number of general delivery from that of intelligence delivery) also increases linearly.

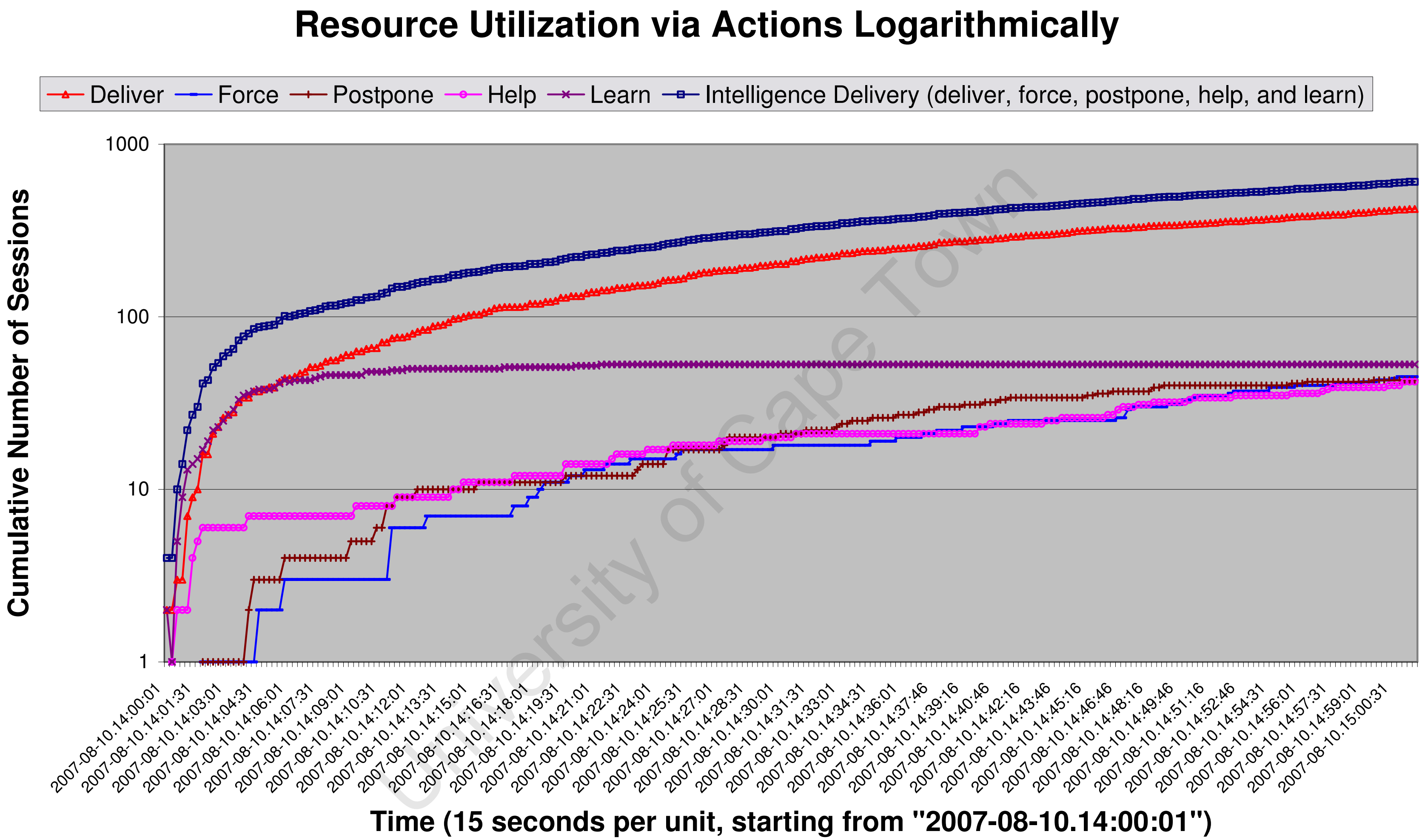


Figure 7-10. Resource utilization through general/intelligence actions over longer term (2).

Y-axis is in logarithmic scale.

We compare intelligence delivery with all its contributors in terms of their ability to facilitate session execution and present their performance logarithmically against linear time in Figure 7-10. It is obvious that all intelligence contributors have a positive effect on session delivery but with different acceleration rates. In the following two paragraphs, we only address the effect of intelligence actions (i.e., the action of *force*, *postpone*, *help*, and *learn*).

At the beginning, the major contribution to intelligence delivery comes from the intelligence actions of *learn* and *help*, especially the session number obtained through the action of *learn* climbing dramatically. These two actions assist in session delivery by utilizing the spare human resource and efficiently exchanging the information resource.

After about four minutes, the actions of *force* and *postpone* appear. After about 10 minutes, they catch up with the previous two intelligence actions in being able to facilitate the session delivery. However, the number affected by the action of *force* almost stops increasing thereafter. This phenomenon indicates that the network has learnt the skill of intelligently handling sessions and thus is able to process sessions by itself with no need to learn any more.

The trends of these numbers for each action are the same if we testify to the system with different types of traffic. Even so, the actual time taken for each number to reach a certain level differs, such as the delayed period for the actions of *force* and *postpone* and the accelerating period for the action of *learn*. These periods are determined by three factors: user social topology, randomly generated traffic, and network decision-making rules. Because the user social topology is fixed and the generated traffic is out of hand, the only amendable factor is the rules.

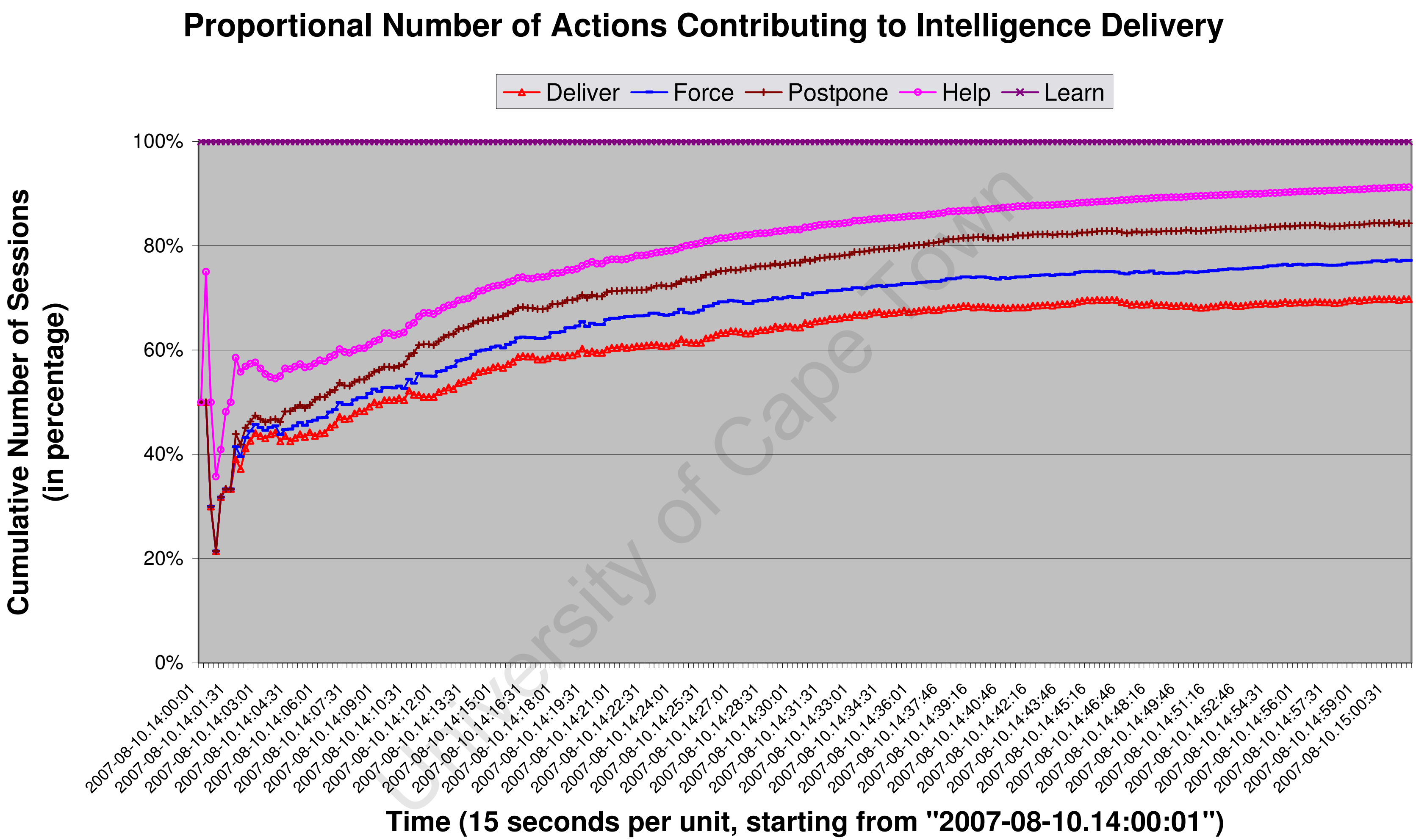


Figure 7-11. Resource utilization through intelligence actions over longer term (3).

All contributors to intelligence delivery expressed by stack lines

Figure 7-11 explicitly shows the proportional contribution of all factors to intelligence delivery while the number of sessions through intelligence delivery itself is not shown. At the beginning, all factors take effort to adapt to the network environment and their contributions to intelligence delivery fluctuate violently. Yet with time, all the contributions approach their respective fixed proportional values. At the end of one hour, among the successfully delivered sessions, 72.63% of them are through the action of *deliver*, 7.75% through *force*, 3.27% through *postpone*, 7.22% through *help*, and 9.12% through *learn*. We can presume from Figure 7-11 that, as time goes on, the proportion of the action *learn* will asymptotically approach zero while the proportions of other actions will stay fixed, provided that the preset decision-making rules do not change. In addition, although there is an increase in the other actions at varying degrees, the number of cumulative sessions through the action of *learn* decreases in proportion with time.

To sum up the sections 7.4.1 and 7.4.2, the virtual-user system helps improve network performance by making effective use of all types of communication resources.

7.5 Enhanced Adaptability to Users' Social Life

The virtual-user system mirrors users' social features in the communications network to improve communication efficiency. Once embedded with the advanced system, the network should have acquired several intelligence abilities of humans, such as the initiatives to learn user social behaviours (section 7.5.1) and the trouble-shooting ability to process temporarily failed sessions according to the changing external environment (section 7.5.2).

7.5.1 Novel Self-Learning Ability

Network self-learning ability is an automatic behaviour of the network to endow a group of users with certain privileges to manipulate their communications. The system embodies this ability by using a set of decision-making rules that regulate the conditions and the actions for the endowing process. This ability functions by copying the communication rules from some users to others based on their common agreement. With this ability, the benefited users save time and energy by the avoidance of repeating what the benefiting users have already done.

Because self-learning factually enables the network to provide an environment where users can learn communication rules from others, the social features of these users play an extremely important role in the learning process. These social features include the users' social relationships, preferences, and social values such as trustworthiness. Figure 7-12 illustrates the number of successfully delivered sessions through the intelligence actions with social features and those without social features, under the communication scenario described in Table 7-6.

Table 7-6. Communication scenario for self-learning ability.

Variable	Unit	Content	Notes
distribution type	N/A	Poisson distribution	In probability, x_0=0, lambda=5
traffic type	N/A	Real-time traffic	E.g., voice call
session-execution start time	SystemTime	2007-08-10.14:00:01	
number of units	N/A	80	
interval of each unit	Second	15	
average number of sessions in each unit	N/A	5	lambda=5
duration of event execution	Second	30	

* x_0, lambda, and N/A: Same as the above "x_0", "lambda", and "N/A".

Table 7-6 provides the information to construct a communication scenario running over 20 minutes, with 80 time units of 15 seconds each. The system generates traffic at the beginning of each unit. The probability of generating a fixed number of session requests at a time abides by the Poisson distribution, with an average of five requests expected in each unit. Figure 7-12 presents the self-learning ability of the network when the network users are connected through Yang's relationship topology (Figure 6-1).

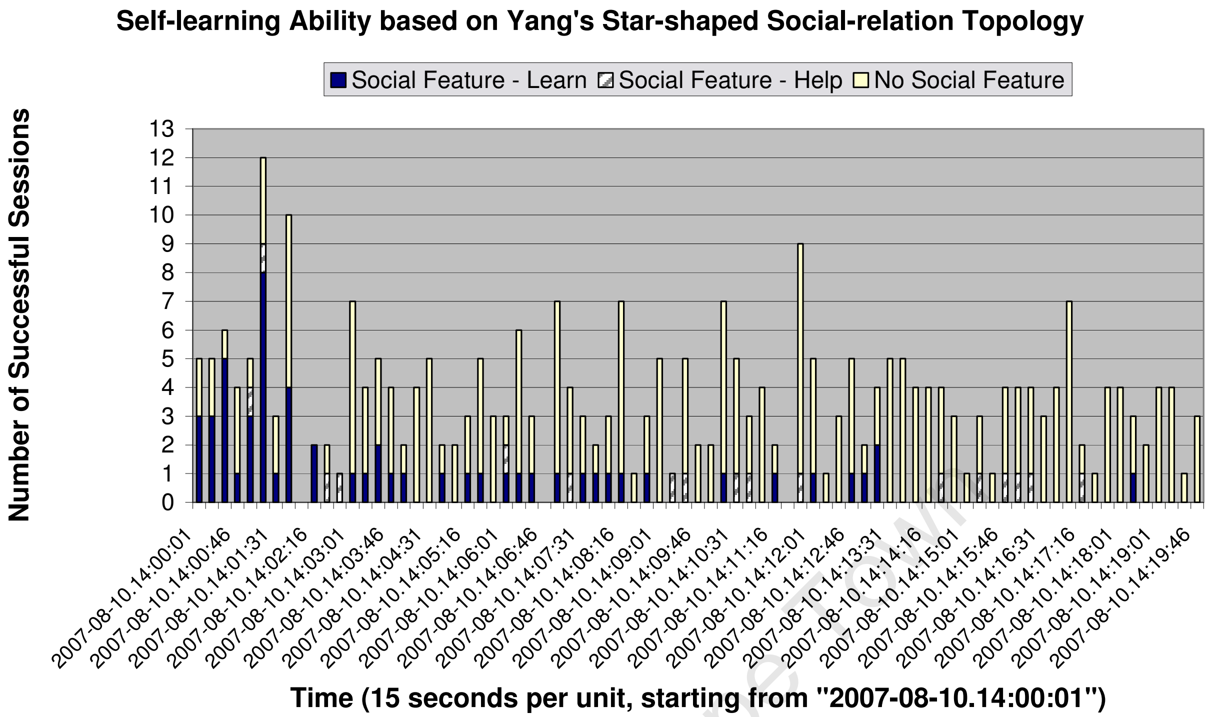


Figure 7-12. Self-learning ability based on Yang's social-relationship topology.

Figure 7-12 depicts the number of sessions delivered through the actions with social features (i.e., the action of *learn* or *help*) and that through the actions without social features.

In the first two minutes of the observation time which we made from the start-up stage of the virtual-user system, the action of *learn* contributes a great deal to session delivery by facilitating a successful session delivery. Meanwhile, the frequency of action occurrence is quite high (almost at every moment), noticeable that we only generate traffic at discrete moments. In the next ten minutes, the system becomes more independent of learning and delivers most sessions using the other non-social manners of *deliver*, *force*, and *postpone*. In the last stage of the observing time, the network is quite capable of handling sessions by itself and only learns when necessary (at the 18th minute). This means that most network users have already learnt the communication rules from others with their permission. The other social-feature related action – *help* – happens whenever needed.

Through the learning process, the intelligence network is able to establish a proper communication environment for the users based on their own social relationships, preferences, and social values. The network is therefore healthier, friendlier, and more convenient.

7.5.2 Dynamic Network-Storage Ability

The dynamic storage ability, also called the dynamic trouble-shooting ability, enables the network to handle session-service requests according to real-time communication environment. It functions via the delivery of normal sessions, the storage of temporarily failed sessions, and the redelivery of these sessions when the exterior environment agrees (sections 4.2.5 and 4.5.2).

We construct a communication scenario using the information provided in Table 7-7 to explicitly exhibit the priority mechanism of the dynamic storage ability in Figure 7-13. An analysis into the figure helps clarify what factors affect the calculation of priority, whether the calculation is able to reflect the communication requirements of the real world, and whether it is able to facilitate a proper process of real-world communication sessions.

Table 7-7. Communication scenario for dynamic storage ability.

Generated Case	Connecting Lifetime	Session Start Time	Emergency Status	Demanding Degree	Application Type	Expected Manner
Deeya -> Phone -> Yang	10 second	14:00:01	Non-emergence	5	Voice call	Deliver
Sam -> Phone -> Yang	10 second	14:00:05	Emergence	4	Voice call	Help
Tao -> Phone -> Yang	10 second	14:00:05	Non-emergence	9	Voice call	Postpone
QiQi -> Phone -> Yang	10 second	14:00:05	Non-emergence	7	Voice call	Postpone
QiQi -> FTP -> Yang	15 second	14:00:05	Non-emergence	7	File Sharing	Postpone

From the information provided in Table 7-7, Deeya phoned Yang at 14:00:01 and prepared Yang's communication status to be busy for 30 seconds thereafter (session execution duration is set to be 30 seconds). After four seconds, Sam, Tao, and QiQi called Yang simultaneously. These users are from different social domains of Yang and thus have various requirements on their call sessions to Yang. The first is an emergency call, the second is a normal call with a high demanding degree of "nine", and the third is a normal call but with a comparatively low demanding degree of "seven". Meanwhile, QiQi also tried to transfer a file to Yang, which contains non-realtime traffic – data.

Yang was still busy when the last four sessions happened, so the network suspended them in the relevant database. Figure 7-13 illustrates the change of priority values for all suspended sessions during their stay in the database.

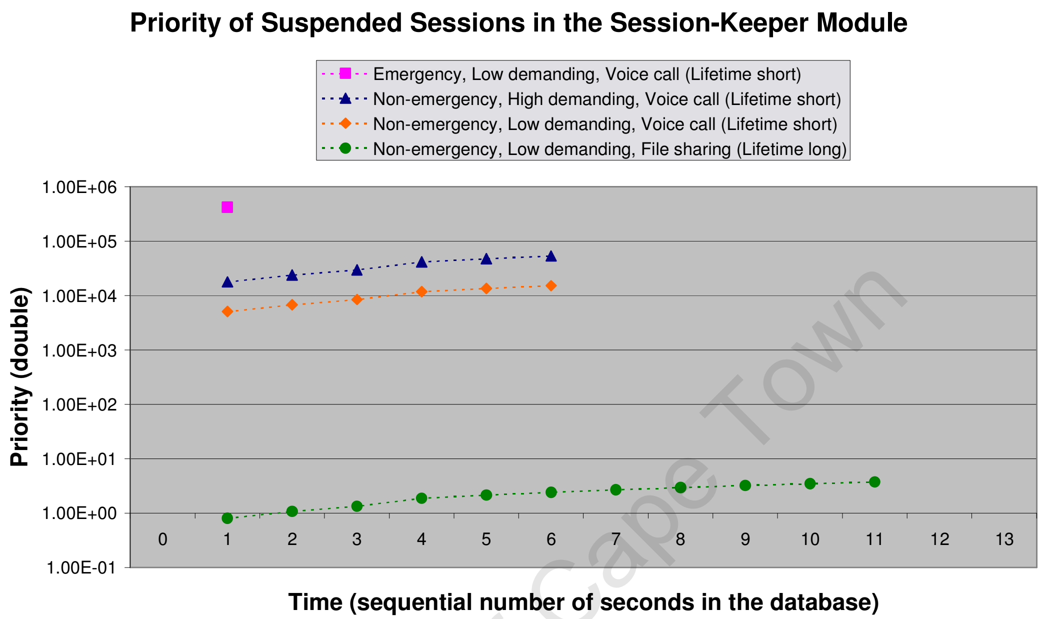


Figure 7-13. Priority of sessions postponed in the Session-Keeper module.

Under the communication scenario described above, Figure 7-13 shows the change of priority value for each suspended session in the Session-Keeper module, with the priority expressed on a logarithmic scale and the priority features regulated in the system (section 4.5.2).

The priority of the session from Sam (marked by rectangles) reached an extremely high value (over 100,000) in a short time due to the feature of emergency. The system, therefore, immediately delivered the session through the action of *help* (referring to Table 7-7) instead of keeping it postponed in the database. Both the session from Tao (marked by triangles) and that from QiQi (marked by diamonds) were normal call sessions. However, Tao had a higher demanding degree to Yang for receiving his call than what QiQi had, so the priority of Tao's call session precedes that of QiQi's call session. Meanwhile, QiQi's file-sharing session (marked by circles) was also stored in the database. The session carried non-realtime data and it therefore possessed an even lower priority than that of QiQi's call session, which carried real-time voice.

Among the four postponed sessions, except Sam's call session that was immediately delivered with the help of other users, the rest had to stay in the database to wait for Yang to become available. However, the three remaining sessions could only survive there until their connecting lifetime expired. The lifetime of the two call sessions expired after the sessions had remained in the database for six seconds and that of the file-sharing sessions expired after staying for 11 seconds. If Yang's status became available in the first six seconds, Tao's call session would be the first choice of the system for processing because it possesses the highest priority. In the next five seconds, the file-sharing sessions would be the only session available for processing because of its long lifetime and the failure of the two call sessions, although it had the lowest priority among the three. Figure 7-13 also implies that the three sessions each had spent around four seconds in previous decision-making processes such as session generating before they reached the database.

7.6 Small Overhead in Time Delay

Despite all the virtues described above, it takes time for the decision-making process to identify an optimal delivery manner. The time taken adds to the delay during the session set-up. Nevertheless, if these delays are within an acceptable limit, the decision-making is still practical.

Under the communication scenario described in Table 7-8, Figure 7-14 shows the time that the system takes to make decisions in an optimal delivery manner for all sessions.

Table 7-8. Communication scenario for time delay in decision-making (short term).

Variable	Unit	Content	Notes
distribution type	N/A	Poisson distribution	In probability, x_0=0, lambda=5
traffic type	N/A	Real-time traffic	E.g., voice call
session-execution start time	SystemTime	2007-08-10.14:00:01	
number of units	N/A	20	
interval of each unit	Second	15	
average number of sessions in each unit	N/A	5	lambda=5
duration of event execution	Second	30	

* x_0, lambda, and N/A: Same as the above "x_0", "lambda", and "N/A".

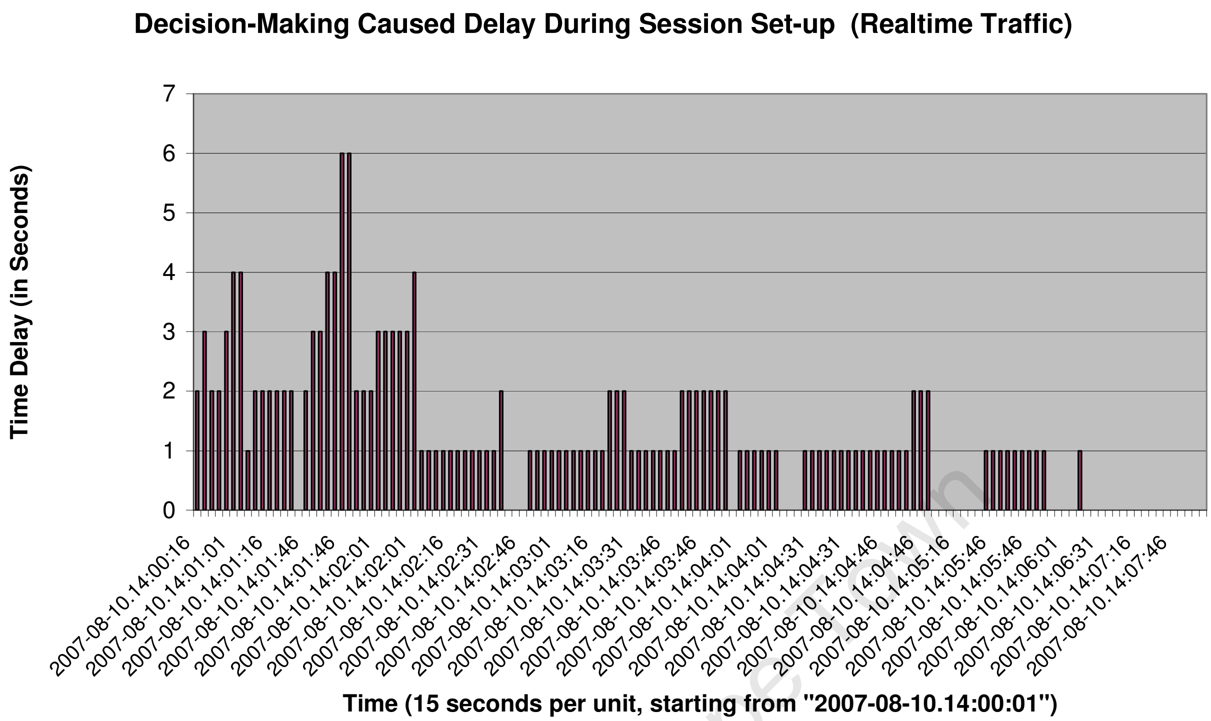


Figure 7-14. Time delay in decision-making for short term.

Figure 7-14 illustrates the time spent in decision-making for the delivery of the sessions that carry real-time traffic. If the intelligence mechanism works for real-time traffic that has the strictest requirement on time delay, it should have no problem for any type of traffic.

According to Figure 7-14, the maximum amount of time consumed in decision-making for the scenario is six seconds, which is acceptable for a voice call. By analyzing the source data of Figure 7-14, we can identify an approximate time range in seconds for all actions as follows: zero to two seconds for the actions of *deliver* and *fail,* one to three seconds for the actions of *force* and *help*, and two to six seconds for the action of *postpone* and *learn*. The statistics data shows that the longest delays come from the actions of *postpone* and *learn.* The delays by the other four actions are between zero and three seconds and are comparably too short to impair system performance or cause uncomfortableness to the users.

Initially, the decision-making tends to make more frequent choices of the *learn* action that generally has a longer delay but is effective in session delivery. Later, the system enhances its intrinsic intelligence ability by having acquired additional delivery manners of *force*, *postpone*, and *help* through learning. When these other manners with smaller delay become more active with time, the system will make less choice of the action of *learn*. Therefore, the proportion that the action of *learn* takes in the total intelligence delivery decreases over time and comes close to zero at the end. At that time, most long delays will mainly come from the delivery by the action of *postpone*.

The time delay caused by decision-making is not harmful to system performance and, therefore, is usable in real-world communication. Even so, for monitoring purposes, it is ideal for the system to be aware of the factors that determine the delay. These affecting factors include the connecting lifetime of the applications that sessions carry, the system sampling-rate, and the session-execution duration. For simplicity purposes, we identify the causes for delay by focusing only on the parameter of the connecting lifetime in Figure 7-15 and Figure 7-16, using the communication scenario described in Table 7-9.

Table 7-9. Communication scenario for time delay in decision-making (long term).

Variable	Unit	Content	Notes
distribution type	N/A	Poisson distribution	In probability, x_0=0; lambda=5
traffic type	N/A	Real-time traffic	E.g., voice call
session-execution start time	SystemTime	2007-08-10.14:00:01	
number of units	N/A	240	
interval of each unit	Second	15	
average number of sessions in each unit	N/A	5	lambda=5
duration of session execution	Second	30	
connecting lifetime	Second	10	For Figure 7-15
		120	For Figure 7-16

* x_0, lambda, and N/A: Same as the above “x_0”, “lambda”, and “N/A”.

Based on the scenario described in Table 7-9, Figure 7-15 illustrates the time delay of all sessions with the allowed 10-second session connecting lifetime.

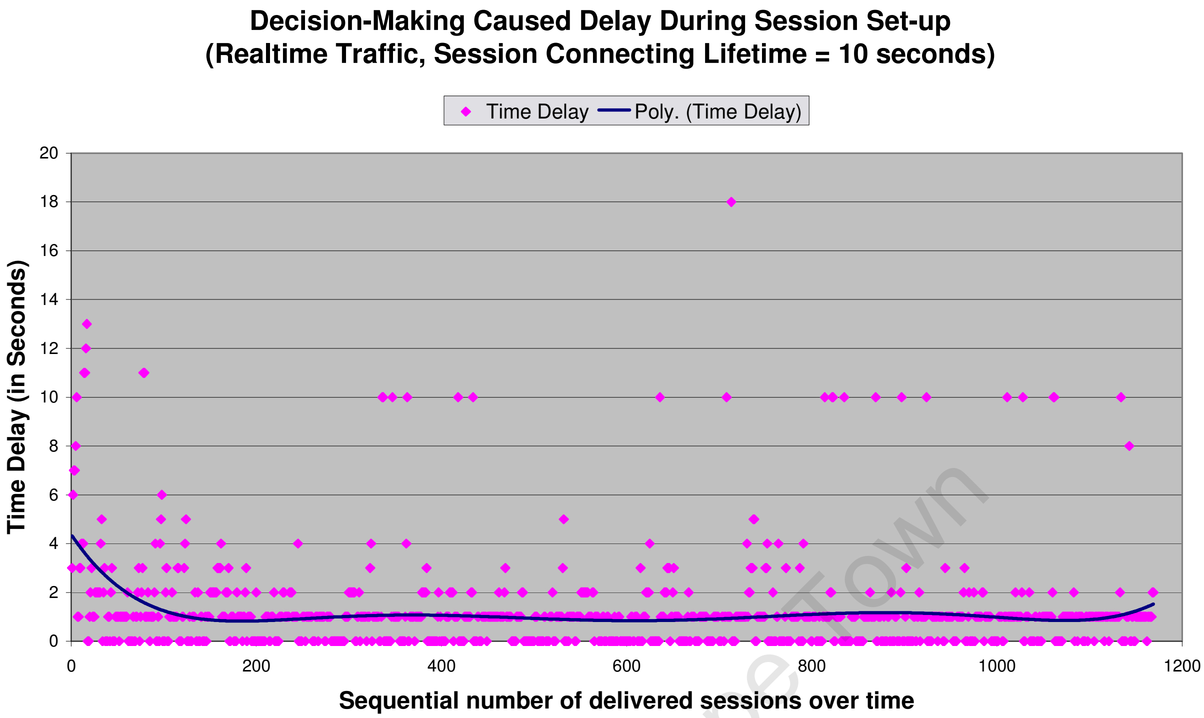


Figure 7-15. Time delay in decision-making for longer term with lifetime of 10 seconds.

Figure 7-15 displays the time delays of sessions executed over one hour, with the session connecting lifetime being 10 seconds. The trend of these delays shows that, although the average delay is high (four seconds) at the beginning due to a massive number of *learn* actions, it develops into a stable value of one second in the end with most long delays only coming from the action of *postpone*. Noticeably, the lifetime of 10 seconds limits the suspending duration of several postponed sessions in the database to 10 seconds. The 18-second delay at around the 710th moment is most probably from the action of *learn* as required. A study of the source data for Figure 7-15 shows that, in the long term, the system immediately delivers 96.58% of the total sessions in the first four seconds with a small portion taking a little bit longer. Only 2.31% of the total sessions are temporarily postponed in Session Keeper for reprocessing.

As has been analyzed, most long-delays in the end are from the action of *postpone*. To exhibit how the session connecting lifetime affects these long delays, we compare the time delay of all sessions in Figure 7-16 where the lifetime is 120 seconds with that in Figure 7-15 where the lifetime is 10 seconds.

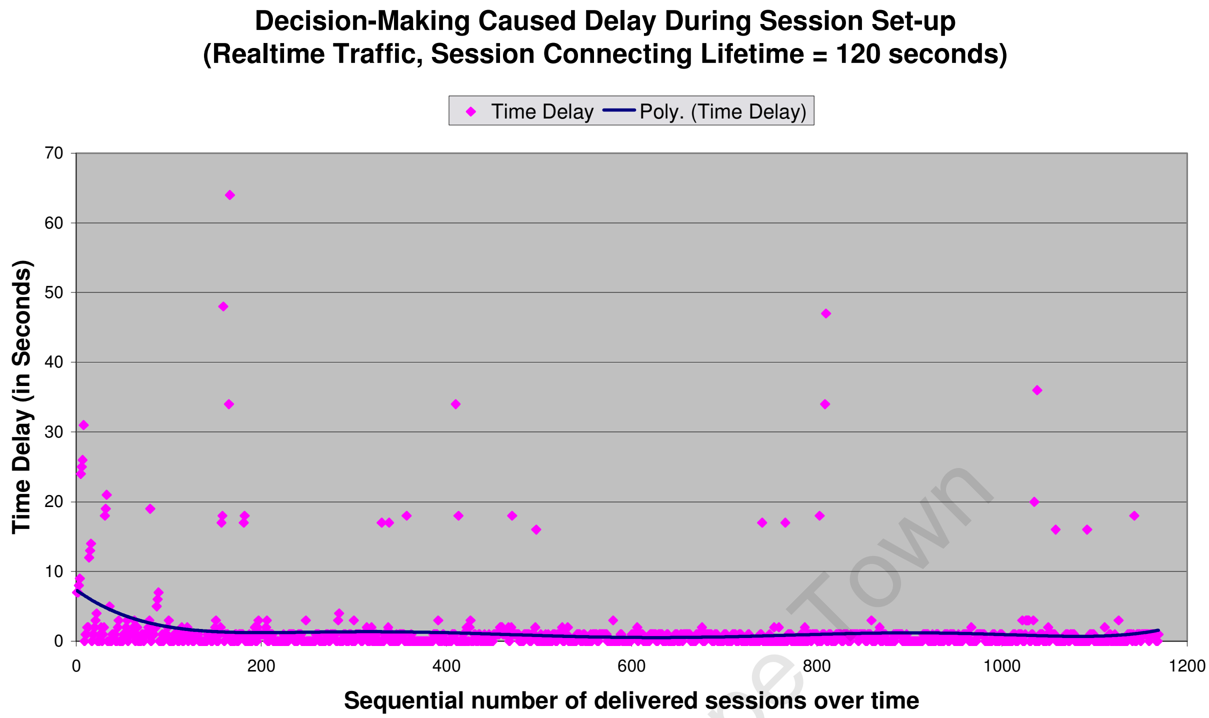


Figure 7-16. Time delay in decision-making for longer term with lifetime of 120 seconds.

Figure 7-16 displays the time delay of sessions executed over one hour, with the connecting lifetime being 120 seconds. To address the generality of the delay, we focus on the delay after the system has become stable in processing the sessions, which are from the 200th moment to the 1170th moment. (1) As in Figure 7-15, the trend of delay in Figure 7-16 also indicates that the delays for most sessions fall in the horizontal belt of delay with a range of zero to four seconds. (2) The second horizontal belt where a second biggest number of sessions reside ranges from 15 to 20 seconds in delay. The belt is formed most probably because of the action of *postpone*. For example, a session started processing at the 200th second. At the 215th second, another session started up and realized that it needed to use the resource occupied by the first session. The second session then had to wait for some time for the first session to release the resource, which would be 30 seconds after the 200th second. At the 230th second, the second session obtained the free resource and started processing 15 seconds later than its starting time, which was the 215th second. Due to other operations such as session generating for decision-making, the first session might take longer – more than 30 seconds – to finish processing, so the second session would get access to the resource in the seconds between 15 and 20. (3) The third horizontal belt locates at around 34 seconds, the fourth at around 48 seconds, and the fifth at around 64 seconds. It is obvious that, the system delivers a number of postponed sessions every 15 seconds on average. The reason is that the system generates traffic every 15 seconds with the session execution time at a constant value of 30 seconds so that the completed sessions can only release network resources for other waiting sessions to use every 15 seconds. For example, the 64-second delay indicates that some session had to wait for four other sessions to release the resources. The reason that it waited so long is because the session had an extremely low priority during waiting time so that it always had to surrender the opportunity of delivery to other sessions. Nevertheless, all these delivered sessions are within the preset connecting lifetime of 120 seconds.

Although some sessions are postponed for longer times (around 2.82% of the total sessions are postponed over 10 seconds), the system still manages to deliver 96.32% of the total sessions in the first four seconds. In reality, system administrators have the flexibility in determining the optimal manner of storing sessions. For example, they can filter the sessions that take too much time in decision-making (e.g., the 18-second delayed session in Figure 7-15 or the 64-second delayed session in Figure 7-16) or adjust the connecting lifetime according to the real-time feature of traffic and user requirements.

7.7 Summary of Chapter 7

This chapter evaluated the performance of the virtual-user system using different parameters. Based on Yang's social-network topology, the system increased the success rate of a general session delivery by 70% at the start-up stage and by 15% in the long term. It greatly balanced network traffic by reducing the traffic fluctuation to around 40% of the original value, regardless of the traffic type. The system helped the network make better use of existing network resources by delivering 23% of the total successful sessions through newly explored resources. It enhanced the network with the advanced intelligence abilities of self-learning and dynamic-storage so that the network is able to behave in a user-friendly manner. Despite these advantages, the system caused a short time delay in the network due to intelligence decision-making.

However, about 96% of the total sessions are delivered within the first four seconds and this short delay is acceptable to the users.

In summary, the evaluation results have proved that the virtual-user system is able to improve communication efficiency in all respects. The improvement is of much value because all these benefits have been gained from the contribution of the users' social features, not from the optimization of the physical network. Besides, the system users will have the flexibility to monitor system performance by adjusting system parameters and modifying preset rules.

Chapter 8 Concluding the Development of Human-like Intelligence Enhanced Network

We first sum up the approach of applying human-like intelligence to the communications network concerning its background, objective, proposal, technical design, and software realization (section 8.1). We then analyze how the approach contributes to network performance and consequently benefits network involvers (section 8.2). Lastly, we present some recommendations for future work (section 8.3).

8.1 Summing Up the Intelligence-Network Approach

The current communications network aims to provide users with a resourceful, convenient, and supporting service environment for information exchange. Yet the aim is hardly achieved when so many of the provided services actually sacrifice users' preferences and desires to accommodate their own drawbacks such as the complexity, inconvenience, and inflexibility in usage. These unfriendly characteristics of services have imposed unnecessary burdens on service users. It is therefore urgent for network designers to regulate the existing services and develop new ones by planning from the user side.

The existing network designs have taken care of users' preferences and desires to a certain extent when considering each user as an individual in his/her involved communication events. However, they have ignored the fact that these preferences and desires could be better achieved if taking into account all communicating users in an event as a whole. In fact, a user is always communicating with another user with whom he/she has a particular relationship. Their relationship determines their mutual views and attitudes and therefore has a far-reaching though indirect effect on the choice of preferred communicating manners. For example, a user may use a low-quality cheap web phone to call his/her families but a high-quality expensive cell phone to call his/her boss, despite undertaking the same type of service – voice call. Unfortunately, current network designs lack an in-depth consideration of such user sociality in establishing communications for users.

With the objectives to satisfy users with personalized services and enable user sociality in the communications network, we hypothesized to import human-like intelligence into the network. Human-like intelligence embodies the human abilities of solving problems according to user preferences and social relations (section 3.2). Enhanced by such intelligence, the network should be able to take the initiative to offer appropriate services to users.

To substantiate the hypothesis, we proposed a virtual-user system to work on behalf of human-like intelligence and implement the system as a newly added component to the network (section 3.3). The system (1) reflects each user's communication life by maintaining his/her personal details, social relationships, and schedules in databases and (2) assists in determining the optimal session-delivery manners using the stored user information. In this way, the system is able to mirror users' community life and intellectual behaviors in the network.

The virtual-user system structurally comprises seven function modules, with each undertaking a specific sub-function towards implementing human-like intelligence (section 4.2). The Session-Generator module prepares an environment and the Session-Comparator module takes the actual actions for identifying communication problems with session processing. The Decision-Maker module follows to solve these problems by determining the most appropriate session-delivery manner according to user requirement. The Session-Keeper module provides storage for the temporarily failed sessions to assist with problem-solving. These four modules make up the core part of realizing human-like intelligence in the network. The Virtual-Personal-Profile module and the Application-Library module respectively provide the user- and application-information for the four previous modules. The Session-Registrar module serves as the interface between the system and the existing physical network. Other than performing specific functions in respective modules, the system applies two important protocols to interwork between these modules (section 4.3). The proactive protocol enables a module to enquire of another module for relevant information to determine its future actions, whereas the reactive protocol allows a module to act on receiving an incoming request from another module.

In addition, the system enrolls nine major mechanisms in its software implementation (section 5.3). (1) The system identifies a user's currently available devices according to his/her preset schedule and the session-processing starting time. (2) The system identifies other potentially practical applications for an application via service type. (3) The system compares two virtual sessions in terms of real-time availability and quality of service. (4) The system determines the optimal session-delivery manner according to the previous comparison results and the user relationship. (5) A procedure of session generating, comparing, decision-making, and decision-executing ensures that the system correctly performs an intelligent session delivery. (6) The system involves trust degree and trustworthiness to facilitate the determination of session-delivery manner. (7) A user learns the information of another user through his/her social relationship. (8) The system expresses the properties of an application as network characteristics, quantifies application performance as a range of value levels for these characteristics, and labels these levels with level IDs. (9) The system searches for the detailed information of applications and users through these level IDs.

The system has proved to be realizable in a software environment. The system modules are not only able to fulfil their respective tasks (section 6.1), but also able to collaboratively implement the manners of *deliver*, *force*, *postpone*, *help*, *learn*, and *fail* for various communication scenarios (section 6.2). Hereof, users' mutual trust degrees and their respective trustworthiness values positively influence the intelligence decision-making on session delivery (section 6.3). The user-friendly GUI and the quantified application performance enrich the system with user-centric properties (section 6.4).

8.2 Contributions of the Research

We have demonstrated that incorporating human-like intelligence and user social-relation into the network (sections 8.2.1 and 8.2.2) is able to provide significant improvement in network performance (section 8.2.3). The development of the intelligence-network approach has also made several minor contributions to general research and community benefits (section 8.2.4).

8.2.1 Novel Idea of Using Human-Like Intelligence to Facilitate Communications

To rescue users from heavy communication burdens, we propose importing human-like intelligence into the network to assist users with their communications. The intelligence comprises users' intellectual abilities to identify occurring communication problems and to construct practical solutions to these problems.

With the human-like intelligence embedded, the network should be able to smartly plan and execute communication sessions according to user requirements. It should also be able to independently perform communication sessions with no need for redundant efforts from users.

To be able to implement abstract human-like intelligence in the network, we have established a virtual user to work on behalf of a real user in his/her communications. The virtual user possesses most intellectual characteristics of the real user by storing his/her communication profile. By referring to the stored user information, the virtual user is able to determine an optimal session-delivery manner that best meets the real user's first desires and provides users a friendly network interface.

Specifically in this project, we mainly focus on how these virtual users handle communications using the real users' social relationship and features. The success of software realization and the achieved improvement in performance have testified to the virtual-user approach.

Another benefit of the virtual user approach is that the virtual user is in fact a set of programming codes and data storage. The network can thus easily interpret abstract user intelligence as easily understood network characteristics in codes.

8.2.2 Feasible Approach for Applying Social Knowledge to Communications

We have successfully established a virtual-user system and testified to the effectiveness of its software implementation (section 8.1). The success of the system proves that applying user intelligence to the real communications network is practical. The system has also proved to be able to assist the network in achieving an outstanding performance in many aspects. Furthermore, it has constructed the solutions to the problems raised earlier (section 1.2.1):

- The two open-source programming languages of Java and MySQL (section 5.1) enable the system to move across different service platforms and environments smoothly.

- The problem-investigating abilities carried by human-like intelligence (section 3.2.3) provide an opportunity for the network to identify the uncertainties in communications, especially those caused by humans (users or network operators).
- The system puts its best efforts forward to deliver services to users through the intelligence decision-making on the optimal session-delivery manner, the types of which include the manners of *deliver*, *force*, *postpone*, *help*, *learn*, and *fail* (section 4.2.4.2).
- By conducting the six best-effort session-delivery manners, the system is making sufficient use of each of the network-, human-, and information-resources.
- The involvement of user social relationships (section 3.3.1) and the thoughtful negotiation mechanisms in decision-making (section 4.2.4.2) collaboratively ensure that the network meets user communication requirements to the utmost, instead of occasionally sacrificing user desires to the encountered technical problems as the current network does.
- These intelligent and sociable virtual users provide a friendly interface for the real users to interact with the network and aptly participate in the real users' communications, consequently saving user labor on unnecessary communications.
- With such user-friendly features and more upcoming, the implemented virtual-user system has proposed a feasible approach for the network operators to continuously attract users by fulfilling their individual requirements on communications to the utmost.

8.2.3 Significant Improvement in Network Performance via User Social Features

The system improves network performance in many aspects through the involvement of network users' social features. According to the analysis (Chapter 7), the system has increased the success rate of communication-session delivery by 16% in the long term and reduced the traffic fluctuation to around 40% of the original value regardless of the traffic type. In addition, the system has helped making better use of existing network resources by delivering slightly more than 23% of the total successful sessions through newly explored resources. Enhanced with the advanced intelligence abilities of self-learning and dynamic-storage via the system, the network is able to behave in a user-friendly manner. Although the system requires a short period for intelligence decision-making, it has still delivered about 96% of the total sessions in the first four seconds to satisfy the users.

The system also improves network security through user sociality. With the adoption of the two-degree trustworthy connection (sections 2.1.2 and 7.1), the system restricts the session initiator to be either an acquaintance or a “known” stranger (i.e., an acquaintance of one of the principal user’s acquaintances). In theory, this restriction can avoid malicious sessions from unknown users.

8.2.4 Other Minor Contributions

Other than improving communication efficiency, the development of the intelligence-network approach has validated the research methodology (section 8.2.4.1) and the experimental results of the approach greatly satisfy different groups of social-communications-network involvers (section 8.2.4.2).

8.2.4.1 Successful Methodology of Importing Social Knowledge into Network

The methodology of implementing the intelligence-network approach develops in accordance with the one illustrated in Figure 2-5. We first identify the unnecessary communication burdens to the users due to the insufficient consideration of users’ social features in service provision (*Identification*). We then propose using human-like intelligence to make up the lack of social features in the network (*Conceptualization*). Thereafter, we design a virtual-user system to represent human-like intelligence and made several interworking mechanisms for the system modules to implement the intelligence (*Formulization*). Lastly, we program this system in software to testify to its functionality (*Implementation*). However, due to the time and experiment-equipment limits, we have not applied the system to real-world platforms or testbeds and therefore cannot use the experimental results as feedback to optimize the system design.

Altogether, the successful establishment of the system has proved that the methodology is functioning and can be widely applied to the bridging of social and communications networks.

8.2.4.2 Achieved Network Ability to Satisfy All Network Involvers

We have successfully designed, implemented, and tested a network approach that is very pleasing to all network involvers[52]. These manipulations on the users coordinate to enhance the intelligence at the network.

The ultimate beneficiary of the intelligence-network approach is the network users. With the assistance of the human-network interaction mechanism, the users do not need to participate in all sessions by themselves. Instead, their virtual users will represent them to organize the best session-delivery manner in terms of appropriate delivery time, optimal delivery manner, suitable receiver, and best price (future work). In addition, the users have flexibility in making their policies on how these virtual users process the sessions according to real users' preferences. They can also update their communication profiles through other communication behaviors such as notifying the system of their most recent physical location by swiping their bank card in a shopping center (future work).

According to the analysis of user communication burdens and their need for social features in communications (section 1.1), service providers are aware of the necessity to provide the users with varied and friendly services and accordingly to introduce services to meet user requirements.

Network operators will gain much from the entire intelligence-network approach. Firstly, the approach has been proven feasible, valid, and ready for use by network operators (Chapter 4 and Chapter 6). Secondly, the study concerning the lack of human factors in the communications network (section 3.1.3) and about what network characteristics are able to carry human nature (section 3.1.4) has generated first-hand material for network operators.

Researchers majoring in networking may be more confident in using the methodology illustrated in Figure 2-5 to apply social knowledge to the communications network. They may

[52] Network involver comprises network user, service provider, network operator, and network researcher.

also be able to identify a wide range of interesting research topics from the cutting edge of social and communications networks (section 8.3).

8.3 Future Work on Reinforcing the Intelligence Design

Despite these significant contributions, the intelligence-network approach needs improving in the areas of system practicability (section 8.3.1), the session-comparison mechanism (section 8.3.2), property of trust-related issues (section 8.3.3), topology of and weights for social relationships (section 8.3.4), security during intelligence session-execution (section 0), and assurance of user privacy (section 8.3.6).

8.3.1 Promote Realizable System to Practical Prototype

This research has validated the theory and software realization of applying human-like intelligence to communications network. These are just the first two stages of bringing the intelligence-network vision into reality – *concept interpretation* and *simulation approval* (section 1.3.3). In these two stages, we used simulated input data and ideal software environment.

In future, we need to prove the practicability of the theory. We can execute emulation traces for the virtual-user system on multi-functional testbed or specific testing tools, using real-world data from social studies as input (*emulation approval*). Based on the emulation results, we then improve the system and test it on the testbed or tools again. After many rounds of such actions of accommodating real-world data and upgrading system performance, we will obtain an optimized software realization of the virtual-user system. Eventually, we are able to release the software as a useable prototype for industry use.

8.3.2 Optimize Reference Points for Comparing Applications

The virtual-user system uses two performance parameters as the reference points to compare the applications carried by the two virtual-sessions in a pair, respectively the real-time availability and the quality of service (section 5.3.3). In fact, the evaluation of application performance depends on more than just these two factors. For example, the factor of billing (e.g., a combination of billing type, cost, and charged party) greatly affects a user's decision-making on whether or not to accept a session request.

In future work, we can involve more parameters to enhance the session comparison. To do so, we first select the parameters relevant to application properties, study their relations to and restrictions on each other, and finally implement the way these parameters determine session comparison in corresponding mechanisms. The more meaningful parameters that the system takes into consideration, the more correct the comparison results are able to interpret user preferences and desires, although the more complex the comparison scheme will be. In addition, various combinations of parameters and numerous ways of combining them may serve the user groups with different session-service requirements.

8.3.3 Employ Dynamic Trust-related Issues

In the virtual-user system, a user's absolute trustworthiness and related trust degrees are static for all sessions in which he/she is involved (section 4.5.1 and Table C-3). However, this static feature is untrue when compared to the user's real-world trust-related natures. In reality, two users' mutual trust degree changes with their interactions when participating in common social activities. In addition, an individual's trustworthiness value changes with his/her social behaviours, problem-solving abilities, and attitudes resulting from the interaction with others.

In future work, we can enable users' trust degree and trustworthiness to dynamically change with the delivery results of each communication session that they participate in, or we can add in further parameters to reflect their changing social features. The adjustable value range of these parameters requires an advanced research in social science.

8.3.4 Construct an Effective User Social Topology

In the virtual-user system, we use Yang's star-shaped topology of social relationships to represent her personal social network, which has less connectedness than in real life (Figure 6-1). We determine whether a user has a social relationship with Yang simply by checking whether that user is in Yang's contact list (section 5.3.7). This oversimplified determination mechanism may exclude from the group of available social users several qualified users who relate to Yang in another way other than via her contact list. Besides, the number of available social domains is limited in the system, e.g., only five domains for Yang's social network (Figure 6-1). In addition, the evaluation of user social relationships is subjective, such as the manually input static trust degree and trustworthiness for a user.

In future, much work is needed in constructing an effective social network for users. This work includes establishing a hierarchical social topology, obtaining relationship information from many resources in users' daily-life behavioural pattern, increasing the number of social domains, and dynamically adjusting the weights that embody users' social features such as trust degree. Lastly, after having implemented the newly built system in the network, we need to forward real-world questionnaires to users to establish whether they approve it or not and, if positive, to what extent they would like to use the social-feature enriched network. We then optimize the system design using these valuable feedbacks.

8.3.5 Analyze the Enhanced Security

To increase the session-delivery success rate to the utmost, we often have to explore new available human resources in the virtual-user system through the intelligence actions of *help* and *learn* (sections 5.3.6, 5.3.7, 6.2.5, and 6.2.6). On the one hand, these two actions may create security problems because they expand the session outside the originally requested trust circle – the original initiator and receiver. On the other hand, they restrict the extended social circle to within the trusted users – the original initiator, the original receiver, and the identified trustable assistant. That is, the system ensures the security for the users by only accepting sessions from a "known" stranger who is still in the principal user's social circle but two degrees away from the user. However, a new problem arises – the system may reject the sessions from a normal user who happens not to be in the two-degree social circle of the principal user.

Future work in enhancing security is to properly define a user's social circle to effectively avoid malicious sessions as well as ensure an adequate accessibility for most normal users. For example, we can extend a user's social circle three degrees away so that more trusted users can communicate with the principal user. Another way is to set up social circles for a user from all aspects of his/her life, such as his/her online shopping activities and banking activities, other than just from communications. These social circles can be based on many social traits of a user such as trust, fidelity, and communicating ability.

8.3.6 Make Flexible Policies for Privacy Purposes

The virtual-user system may cause intrusion into the initiator's privacy when applying the action of *help* (section 6.2.5), although we have attempted to reduce this side effect to a minimum using a set of negotiation mechanisms (section 4.2.4.2) and trust-related policies (sections 5.3.6 and 5.3.7).

Enhancing user privacy will be one of the major concerns in future work. For example, we can make several flexible policies for different privacy requirements so that the users have the flexibility to choose their preferred level of privacy for communication sessions. We can also optimize trust performance or other human nature related performance of the system so that it can suggest the most suitable third parties to the users. Lastly yet importantly, we need to convince users to accept a communications network enhanced with their own social networks, because the social-communications-network is able to provide satisfying convenience, improved quality of service, better security, ensured privacy, and complete user-friendly features.

References

[1] S. Segaller, *Nerds 2.0.1: A brief history of the Internet*, TV Books, 1st ed., New York, United States, 1998. ISBN: ISBN: 1-575-00106-3.

[2] J. F. Kurose and K. W. Ross, *Computer networking: A top-down approach featuring the Internet*, Pearson Education Inc., 3rd ed., United States, 2005. ISBN: 0-321-26976-4.

[3] International Telecommunication Union, "Living the Digital World", *The Forum At ITU Telcom World 2006*, Hong Kong, Dec. 2006. Available: http://www.itu.int/WORLD2006/forum/ .

[4] Vonage, *homepage*. Available: http://www.vonage.com/index.php?ic=1 .

[5] B. Briscoe, A. Odlyzko, and B. Tilly, "Metcalfe's Law is Wrong," *IEEE Spectrum*. Available: http://spectrum.ieee.org/print/4109 .

[6] C. Daniel, M. Bick, R. Egan, K. Mills, S. Walker, and S. Furukawa, "MSF R2 service architecture," *MSF technical report*, Jun. 2003. Available: http://www.msforum.org .

[7] H. Gardner, *Frames of mind: The theory of multiple intelligences*, BasicBooks (a division of HarperCollins Publishers), 1983. ISBN 0-465-02510-2.

[8] D. Wechsler, *The measurement of adult intelligence*, The Williams & Wilkins Com., 3rd ed., Baltimore, United States, 1994.

[9] M. A. Rothenberg, D. A. Foster, and S. Žižek, *Perversion and the Social Relation*, Duke University Press, 2003. ISBN: 0-822-33097-0.

[10] R. Westerveld and R. Prasad, "Rural communications in India using fixed cellular radio systems," *IEEE Communications Magazine*, vol. 32, no. 10, pp. 70-77, Oct. 1994.

[11] L. LefEver, "Video: Social Networking in Plain English," Web pages. Available: http://www.commoncraft.com/shows/socialnetworking .

[12] The Conference Board of Canada, "Innovation skills profile," *webpage*. Available: http://www.conferenceboard.ca/education/learning-tools/isp.htm .

[13] Elizabeth F. Churchill and Christine A. Halverson, "Social networks and social networking," *IEEE Internet Computing Magazine*, vol. 9, no. 5, 2005, pp. 14-19.

[14] J. Scott, *Social Network Analysis: A handbook*, 2nd ed., Sage Publications, 1991.

[15] M. Granovetter, "The strength of weak ties," *American J. Sociology*, vol. 78, no. 6, pp. 1360–1380, May 1973.

[16] L. C. Freeman, *The development of Social Network Analysis: A study in the sociology of science*, Empirical Press, Vancouver, 2004. ISBN: 1-594-57714-5.

[17] M. E. J. Newman, “The mathematics of networks,” *web archive*, Center for the Study of Complex Systems, University of Michigan, Michigan, United States. Available: http://www-personal.umich.edu/~mejn/papers/palgrave.pdf .

[18] Facebook Comp., *homepage*. Available: http://www.facebook.com/ .

[19] Orkut, *homepage*. Available: www.orkut.com .

[20] A. L. Barabasi, *Linked: How everything is connected to everything else and what it means*, Penguin Group Inc., New York, May 2003. ISBN: 0-738-20667-9.

[21] J. Kleinberg, “The small-world phenomenon: an algorithm perspective,” *Proc. 32nd annual ACM symposium on Theory of Computing*, Portland, United States, 2000, pp. 163–170. ISBN: 1-581-13184-4.

[22] S. Milgram, “Behavioral study of obedience,” *J. Abnormal and Social Psychology*, iss. 67, pp. 371–378, 1963.

[23] LinkedIn Corp., *homepage*. Available: http://www.linkedin.com/ .

[24] FON. Available: http://www.fon.com/en/ .

[25] Boingo. Available: http://www.boingo.com/ .

[26] M. Weiser, “The Computer for the 21st Century,” *ACM SIGMOBILE Mobile Computing and Communications Review*, Vol. 3, Iss. 3, pp. 3-11, 1999.

[27] M. Weiser, R. Gold, and J. S. Brown, “The origins of ubiquitous computing research at PARC in the late 1980s,” *IBM J. Systems*, vol. 36, no. 4, pp.693-696, 1999.

[28] E. de Lara and M. Ebling, “New products,” *IEEE Mag. Persvasive Computing*, pp. 11-13, Jan. – Mar., 2007.

[29] R. Want, “You are your cell phone,” *IEEE Mag. Persvasive Computing*, pp. 2-4, Apr. – Jun., 2008.

[30] D. A. Bowman and R. P. McMahan, “Virtual Reality: How much immersion is enough?” *IEEE Mag. Computer*, pp. 36-43, Jul. 2007.

[31] Second Life, “Create an avatar,” *webpage*. Available: http://secondlife.com/whatis/avatar.php .

[32] M. N. Kamel Boulos, L. Hetherington, and S. Wheeler, “Second Life: an overview of the potential of 3-D virtual worlds in medical and health education,” *J. Health Information and Libraries 24 (4)*, vol. 24, iss. 4, pp. 233-245, Dec. 2007.

[33] J. McCarthy, “What is Artificial Intelligence – Basic questions,” *webpage*, Computer Science Department, Stanford University, United States, Nov. 2004. Available: http://www-formal.stanford.edu/jmc/whatisai/node1.html .

[34] A. Newell, "AI Overview," *AAAI press*. Available: http://www.aaai.org/AITopics/html/overview.html .

[35] E. Feigenbaum, P. McCorduck, and H. P. Nii, "Expert Systems," *web archive.* Available: http://www.aaai.org/AITopics/html/expert.html .

[36] *The American Heritage Dictionary of the English Language*, Houghton Mifflin Comp., 4th ed., Boston, United States, 2006. ISBN: 0-618-70172-9.

[37] International Business Machines Corp., *homepage*. Available: http://www.ibm.com/us/ .

[38] International Business Machines Corp., "Kasparov vs. Deep Blue," *web archive*. Available: http://www.research.ibm.com/deepblue/home/html/b.shtml .

[39] Association for the Advancement of Artificial Intelligence, "Regarding the Intelligence in Distributed Intelligent Systems," *AAAI Fall Symposium Series Call For Paper*, 2007. Available: http://www.aaai.org/Symposia/Fall/fss07.php .

[40] D. Partridge, *Artificial Intelligence and Software Engineering – Understanding the promise of the future*, Routledge, United States, Jan. 1998. ISBN: 1-579-58062-9.

[41] F. Hayes-Roth, D. A. Waterman, and D. B. Lenat, *Building Expert Systems*, Addison-Wesley Longman Publishing Co., Inc. Boston, United States, 1983. ISBN: 0-201-10686-8.

[42] S. Franklin and A. Graesser, "Is it an Agent, or just a Program? A Taxonomy for Autonomous Agents," *Proc. 3rd Int'l Workshop on Agent Theories, Architectures, and Languages*, Springer-Verlag, 1996.

[43] L. Foner. "Entertaining Agents: A sociological case study," *Proc. 1st Int'l Conf. Autonomous Agents (Agents'97)*, Marina del Rey, Canada. Available: http://foner.www.media.mit.edu/people/foner/Julia/Julia.html

[44] S. -Y. Choi, D. O. Stahl, and A. B. Whinston, *The economics of electronic commerce*, MacMillan Publishing Co., 1st ed., 1997. ISBN: 1-578-70014-0. Available: http://crec.mccombs.utexas.edu/works/ebook/ec01.pdf .

[45] U. W. Birchler and M. Bütler, *Information economics*, Routledge, 1st ed., London, Jul. 2007. ISBN: 0-415-37346-8.

[46] IBM T. J. Watson Research, "Intelligent Agents project at IBM T. J. Watson Research," 1994-1997. Available: http://www.research.ibm.com/iagents/ibm_iagents.html .

[47] Sh. M. Wu, H. Ghenniwa, Y. Zhang, and W. M. Shen, "Personal assistant agents for collaborative design environments," *Special Issue: Collaborative environments for concurrent engineering - Archive of Computers in Industry*, vol. 57, iss. 8, Dec. 2006, pp. 732 - 739. ISSN: 0166-3615.

[48] D. J. Watts, *Six Degrees: The science of a connected age*, W. W. Norton & Company, 1st ed., Feb. 2003. ISBN: 0-393-32542-3.

[49] Info. Sciences Institute and Univ. of Southern California, "Internet protocol – DARPA Internet Program," *IETF RFC 791*, Sep. 1881. Available: http://tools.ietf.org/rfc/rfc791.txt .

[50] Deutsche Telekom, "Glossary: Public switched telephone network (PSTN)," *group report*. Jan. – Mar., 2007. Available: http://www.zwischenbericht.telekom.de/site0107/en/co/glossar/index.php .

[51] Advanced Relay Corporation, "LayGO user's guide: Packet switched public data network," *webpage*, Jan. 2007. Available: http://www.advancedrelay.com/laygodoc/laygousr/index.htm?http://www.advancedrelay.com/laygodoc/laygousr/appendb.htm .

[52] ITU-T REC I.120, "Integrated Service Digital Network (ISDN): General structure," Mar. 1993.

[53] International Telecommunication Union, homepage. Available: http://www.itu.int/net/home/index.aspx .

[54] CCITT I.312/Q.1201, "Principles of intelligent network architecture," 1993.

[55] ITU-T REC Q.1211, "Introduction to Intelligent Network Capability Set-1," Mar. 1993.

[56] ITU-T REC Q.1221, "Introduction to Intelligent Network Capability Set-2," Sep. 1997.

[57] ITU-T REC Q.1231, "Introduction to Intelligent Network Capability Set-3," Dec. 1999.

[58] ITU-T REC Q.1241, "Introduction to Intelligent Network Capability Set-4," Jul. 2001.

[59] ITU-T REC H.323, "Packet-based multimedia communications systems," Jun. 2006.

[60] GSM World, *homepage*. Available: http://www.gsmworld.com/index.shtml .

[61] 3GPP TS 23.060, "General Packet Radio Service (GPRS) – Service description," stage 2, ver. 7.5.0, Sep. 2007.

[62] 3GPP TS 23.078, "Customized applications for mobile network enhanced logic (CAMEL) phase 4," stage 2, ver. 7.0.0, Jun. 2005.

[63] T. Magedanz, "Intelligent network (IN) in wireless – 3GPP CAMEL," *lecture notes*, Aug. 2005.

[64] Federal Standard 1037C, "Telecommunications: Glossary of Telecommunication Terms," *define: public land mobile network (PLMN)*, Aug. 1996. Available: http://www.its.bldrdoc.gov/fs-1037/fs-1037c.htm .

[65] 3GPP TS 03.40, "Technical realization of the Short Message Service (SMS)." Available: http://www.3gpp.org/ftp/Specs/html-info/0340.htm .

[66] American National Standards Institute, *homepage*. Available: http://www.ansi.org/ .

[67] Telecommunications Industry Association, *homepage*. Available: http://www.tiaonline.org/ .

[68] The Internet Engineering Task Force, *homepage*. Available: http://www.ietf.org/ .

[69] M. Poikselka, G. Mayer, H. Khartabil, and A. Niemi, *The IMS: IP Multimedia Concepts and Services*, John Wiley & Sons, Ltd., 2nd ed., 2006. ISBN: 0-470-01906-9.

[70] 3GPP TS 23.228, "IP Multimedia Subsystem (IMS)," stage 2, ver. 7.1.0, Sep. 2005. Available: http://www.3gpp.org/specs/specs.htm .

[71] 3GPP2 X.S0013-000-0, "All-IP core network multimedia domain – Overview," ver. 2.0, Aug. 2005.

[72] F Kon, F Costa, G Blair, and R. H. Campbell, "The case for reflective middleware", *Communications of the ACM, Special Issue: Adaptive middleware*, vol. 45, iss. 6, pp. 33-38, Jun. 2002. ISSN: 0001-0782.

[73] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler, "SIP: Session Initiation Protocol," *IETF RFC 3261*, Jun. 2002. Available: http://www.ietf.org/rfc/rfc3261.txt .

[74] International Telecommunication Union, "ITU-T's Definition of NGN", *web archive*, 2007. Available: http://www.itu.int/ITU-T/ngn/definition.html .

[75] ITU-T REC E.800, "Terms and definitions related to quality of service and network performance including dependability," Aug. 1994. Available: http://www.itu.int/rec/T-REC-E.800-199408-I/en .

[76] ITU-T REC X.641, "Information technology – Quality of Service: Framework," Dec. 1997. Available: http://www.itu.int/rec/T-REC-X.641-199712-I/en .

[77] B. Moore, "Some comments on ITU-T work on NGN," *EU Commission Open Workshop on Identifying policy and regulatory issues of NGN*, Brussels, Jun. 2005.

[78] V. K. Garg, *Wireless Network evolution: 2G to 3G*, Prentice Hall PTR, Aug. 2001. ISBN: 0-130-28077-1.

[79] Yahoo Inc., Yahoo Identification, *homepage*. Available: http://www.yahoo.com .

[80] 3G Forum, "Just joined Vodafone, major concerns," Aug. 2006. Available: http://www.3g.co.uk/3GForum/archive/index.php/t-44344.html .

[81] Y. Li and H. A. Chan, "A Mechanism of applying intelligence to the network by considering users' social relations," *Proc. IEEE 13th Int'l Conf. Telecommunications (ICT'06)*, Madeira, Portugal, May 2006.

[82] Y. Li and H. A. Chan, "Dynamic service-session database to enhance service process in communication networks," *Proc. IEEE Int'l Conf. AFRICON'07*, Windhoek, Namibia, Sep. 2007.

[83] M. Sipser, *Introduction to the theory of Computation*, Thomson Course Technology, 2nd ed., 2006. ISBN-10: 0-534-95097-3.

[84] Y. Li and H. A. Chan, "A trust scheme of Intelligence Networks – for service delivery," *Proc. Conf. SATN/AC'06*, Cape Town, South Africa, Sep. 2006. ISBN: 0-620-37043-2.

[85] T. Magedanz, D. Witaszek, and K. Knuettel, "The IMS Playground @ Fokus - An open testbed for Next Generation Network multimedia services," *Proc. 1st Int'l Conf. Testbeds and Research Infrastructure for the Development NeTworks and COMmunities 2005 (TRIDENTCOM'05)*, pp. 2-11.

[86] Sun Microsystems Inc., Java, *webpage*. Available: http://www.java.com/en/about/ .

[87] Sun Microsystems Inc., NetBeans, *webpage*. Available: http://www.netbeans.org/ .

[88] MySQL AB, MySQL, *webpage*. Available: http://www.mysql.com/ .

[89] R. Ramakrishnan and J. Gehrke, *Database management systems*, McGraw-Hill Professional, 2003. ISBN: 0-072-46563-8.

[90] Serena, SQL, *webpage*. Available: http://www.sql.org/ .

[91] Sun Microsystems Inc., "JDBC: JDBC overview," *webpage archive*. Available: http://java.sun.com/products/jdbc/overview.html .

[92] MySQL AB, "Download Connector/J 5.0," *webpage*. Available: http://dev.mysql.com/downloads/connector/j/5.0.html .

[93] MySQL AB, "MySQL GUI tools downloads," *webpage*. Available: http://dev.mysql.com/downloads/gui-tools/5.0.html .

[94] Sun Microsystems Inc., "NetBeansTM IDE 5.0 release notes," *webpage*. Available: http://www.netbeans.org/community/releases/50/relnotes.html .

[95] MySQL AB, "MySQL server FAQ," *webpage*. Available: http://www.mysql.com/industry/faq/ .

[96] D. Nourie, "Getting started with an Integrated Development Environment (IDE)," *webpage: Sun Microsystems Inc.*, Mar. 2005. Available: http://java.sun.com/developer/technicalArticles/tools/intro.html .

[97] R. Hinden and S. Deering, "IP Version 6 addressing architecture", *IETF RFC 2373*, 1884. Available: http://www.ietf.org/rfc/rfc2373.txt .

[98] J. Lewis, W. Loftus, and C. Cocking, *Java Software Solutions*, Addison Wesley Longman, 1st ed., Jan. 2003. ISBN: 0-201-88259-0.

[99] T. H. Cormen, Ch. E. Leiserson, and R. L. Rivest, *Introduction to Algorithms*, MIT Press and McGraw-Hill, 1st ed., 1990. ISBN: 0-262-03141-8.

[100] M. Handley and V. Jacobson. "SDP: Session Description Protocol," *IETF RFC 2327*, Apr. 1998. Available: http://www.ietf.org/rfc/rfc2327.txt .

[101] A. Voisard, "Towards a toolbox for Geographic User Interfaces," *Proc. 2nd Int'l Symposium on Advances in Spatial Databases*, London, United Kingdom, 1991, pp. 75 – 97. ISBN: 3-540-54414-3.

[102] F. A. Haight, *Handbook of the Poisson Distribution*, Wiley, Jun. 1967. ISBN: 0471339326.

[103] A. K. Kaw, *Introduction to matrix algebra*. University of South Florida, 2002. Available: http://numericalmethods.eng.usf.edu/matrixalgebrabook/downloadma/index.html .

[104] 3rd Generation Partnership Project, *homepage*. Available: http://www.3gpp.org/ .

[105] 3rd Generation Partnership Project 2, *homepage*. Available: http://www.3gpp2.org/ .

[106] ITU-T REC E.168.1, "Assignment procedures for universal personal telecommunications (UPT) numbers in the provisioning of the international UPT service," Feb. 2005.

[107] P. McGregor, R. Kaczmarek, V. Mosley, D. Dease, and P. Adams, "National security/Emergency preparedness and the Next-Generation Network," *IEEE Communications Magazine*, vol. 44, no. 5, pp. 133-143, May 2006.

[108] Y. H. Tan and W. Thoen, "Formal aspects of a generic model of trust for electronic commerce," *Proc. IEEE 33rd Hawaii Int'l Conf. System Sciences*, Jan. 2000.

[109] X. F. Zhang and Q. Zhang, "Online trust forming mechanism: Approaches and an integrated model," *Proc. ACM 7th Int'l Conf. Electronic Commerce (ICEC'05)*, Xi'an, China, Aug. 2005, pp. 201-209.

[110] D. Gefen, "Reflections on the dimensions of trust and trustworthiness among online consumers," *ACM SIGMIS Database for Advances in Information Systems*, vol. 33, iss. 3, 2002, pp. 38-53.

[111] S. P. Marsh, "Formalizing trust as a computational concept," *PhD dissertation*, Dept. Computing Science and Mathematics, Stirling Univ., Scotland, United Kingdom, 1994.

[112] E. J. Chang, F. K. Hussain, and T. S. Dillon, "Fuzzy nature of trust and dynamic trust modeling in service oriented environments," *Proc. Conf. ACM Secure Web Services (SWS'05)*, Fairfax, United States, Nov. 2005, pp. 75-83.

[113] S. W. Lee, R. A. Gandhi, and G. J. Ahn, "Establishing trustworthiness in services of the Critical Infrastructure through certification and accreditation," *Proc. Conf. ACM Software Engineering for Secure Systems – Building Trustworthy Application (SESS'05)*, St. Louis, United States, May 2005.

[114] M. H. Zhou, W. P. Jiao, and H. Mei, "Customizable framework for managing trusted components deployed on middleware," *Proc. IEEE 10th Int'l Conf. Engineering of Complex Computer Systems (ICECCS'05)*, ShangHai, China, Jun. 2005, pp. 283-291.

Appendix A Explanations and Examples to Statements

A.1 Examples for the problem statement in section 1.2.1

The advantages for the network to process human perspectives are briefed in section 1.2.1 and detailed examples are given below.

(1) Practical user-centric service platform that is deliberately developed for seamless use under different network vendors [6] has plenty of room for improvement. An individual current service platform such as the Intelligent Network Capability Set 1/2/3/4 under fixed networks [54]-[58] or the Customized Applications for Mobile network Enhanced Logic under wireless networks [62]-[63] only serves each vendor's individual network one at a time. The IP Multimedia Subsystem under the standardization organizations of Third Generation Partnership Project (3GPP[53] [104]) and Third Generation Partnership Project 2 (3GPP2[54] [105]) aims to work under multiple vendors is currently only at the beginning stage of deployment [70][71][85].

(2) Current communications network will be able to detect several man-made uncertainties and therefore offer proper solutions to reduce the damage to communication events caused by these uncertainties. One example of the man-made uncertainties is that network operators may deliberately make the policy that low-priority events surrender to higher-priority events under specific circumstances. Hereof the priority value depends on the content of the service. Another example is that network users may not be personally available for an event at

[53] 3GPP is collaboration between groups of telecommunication associations that produce globally applicable Technical Specifications and Technical Reports for Third Generation mobile system based on the evolved core networks of Global System for Mobile communication (GSM) and the radio access technologies of Universal Terrestrial Radio Access, General Packet Radio Service, and Enhanced Data rates for GSM Evolution.

[54] 3GPP2 is collaboration between telecommunications associations to make a globally applicable Third Generation mobile phone system specification within the scope of the International Telecommunication Union's International Mobile Telecommunications "IMT-2000" project.

the service execution moment. These network-involver-caused uncertainties generally cause the failure of a communication event early in its set-up stage.

(3) Current communications network will be able to deliver better service to users if taking the initiatives to identify potential connecting manners for users. However, these initiatives are lacking in current networks. First, although the network possesses full connectivity of a user, it does not take the initiatives to enumerate all his/her possible connecting manners to get hold of him/her. For example, when executing call-forwarding service, the network only follows the forwarding rules that were manually preset by users. It would not bother that users may have missed several connecting possibilities. Second, the network does not put efforts to identify optimal, even potential manners for users. For example, the network stores Lisa's office-phone number "123" and cell number "101", and is aware of the number "999" of a public phone through which Lisa just made a call. When Bill tries to phone Lisa, any of the following three scenarios may occur. In the most common scenario, Bill memorizes Lisa's two personal phone numbers and manually dials Lisa's office-phone number then cell number. In a more advanced scenario where Universal Personal Telecommunications ([106]) fit in, Bill dials a universal identification number for Lisa and the network connects to Lisa's office phone and cell in sequence according to Lisa's preset call-forwarding rule. Yet in the last scenario where both Lisa's personal phones are unreachable, Bill has no way to reach Lisa because the network would not proactively identify the public phone number "999" and propose the number to Bill.

(4) Current communications will be able to make more efficient use of each of the network-, human-, and information-resources and create a mechanism to interact between them. Nowadays, when Lisa is busy on her phone (both the network and the human resources about Lisa are busy) and Bill is not using his phone (the network and the human resources for Bill are free), the network normally fails the general-query call that comes in for Lisa. However, if the network is able to direct the call to Bill, it will make sufficient use of all types of communication resources[55]. The free network- and human-resources will be used (Bill and his phone), the busy

[55] Communication resources comprise network-, human-, and information-resources.

network- and human-resources will not be overloaded (Lisa and her phone), and the information resources get spread rather than stopped (the call is answered). Note that, although some existent services such as call-center service allow the reallocation of a call to the next available agent, for whom the service heads is based on the predefined rules. The network is not smart enough to diagnose the attributes and contents of the call and then divert the call to the most appropriate receiver according to user profile. For example, if Lisa is not immediately available for an incoming call, the current network is not able to make the decision of diverting the call to Bill if the call is from a general customer and to her boss if the call is from an important customer. All in one, a reasonable collaborative use of these communication resources will improve communication efficiency via accessing each others' broader resources and will save on both production means (network resources) and productivity (human labor).

(5) Current network can avoid sacrificing users' first desire to solve the technical problems encountered. An example of such sacrifice is that the network operators use the communication policies of National Security/Emergency Preparedness (NS/EP) to block several normal services that have started but not got through to the receiver in order to connect governmental or highly confidential services during peak hour [107]. However, this policy of dropping normal services is unfair to service users.

(6) The wasted resources of communication society, in which thousands of users are repeating the same operation steps in order to use the same service, can be reduced. This is because the users who are using the same type of services actually share certain social relations. Examples of these social relations include performing the same task in office, partaking in the same leisure activity in a club, or sharing the moments of happiness as well as sadness together at home. Thus, the current communications network may seriously consider these social relationships between the users and then employ these relationships to help network users with their communications.

(7) In failure analysis and corrective actions for unsuccessful communication events, most studies have been focusing on the technical problems arising from the physical network such as congested traffic and information protection. It is desired to extend these studies to

include the unavailability of desired communicating parties (e.g., expected session receiver) or the underdeveloped potential distribution of human resources over the network.

(8) Customer satisfaction can be increased. Network operators may lose current customers if the operators cannot meet users' individual requirements on service. User-centric service-provision platforms enables the network to attract and enroll new customers in exploring new types of service under different access networks.

A.2 Conducted research on trust for section 2.1.3

In section 2.1.3, we stated that current researches on trust have provided a method of quantifying trustworthiness and trust degree by following the steps below. Hereof specific researches are appended to each step. (1) Analyze the generic concepts of trust degree and trustworthiness. For example, Tan and Thoen proposed assigning each network node a trust rating based on its performance in the network [108] and Zhang advised three approaches of "cycle," "stage development," and "factor" to manipulate these trust ratings [109]. (2) Reinforce the basic concepts of trust degree and trustworthiness by considering as many causal factors of them as possible. For example, Gefen brought forward an advanced three-dimensional scale of trustworthiness, including integrity, benevolence, and ability, versus the popular one-dimensional one [110]. (3) Formulate the trust degree and the trustworthiness. For example, Marsh formulated how an agent can trust another by considering their temporal constraints, different situations and their similarities, their past behaviours, and the environments [111]. (4) Select suitable modelling languages for the implementation of the formulae. For example, Chang presented a pictorial language to model the dynamic nature of trust [112]. (5) Simulate trust degree and trustworthiness in software. For example, Lee proposed a software tool with a well-structured framework to establish levels of trustworthiness for services to help with information processing [113]. For example, Zhou etc. further came up with a complex middleware to protect and evaluate the trust-related issues [114]. Joining the above research methodologies into a job sequence enables us to define and calculate trust degree and trustworthiness.

Appendix B Fields of Tables in Information Databases

The information provided in Table B-1 shows the data types in MySQL:

Table B-1. Data types used for MySQL information databases.

Data Type	Explanation
VARCHAR(*n*)	The type of data is String, with a length of *n* characters.
INTEGER	The type of data is Integer.
ENUM(*string*)	The type of data is Enumeration, with the enumerated values expressed in *string*.

B.1 User Database for the Virtual-Personal-Profile Module

The information provided in Table B-2 to Table B-5 shows users' communication profile.

Table B-2. Table "user_index" stores key information for all users.

Field	Type	Feature	Explanation	Notes	Example
id	INTEGER	Primary key	Record index		1
user_id	VARCHAR(4)	Foreign key	User identification	Unique over network	AA11
name	VARCHAR(25)		User name		Yang
trustworthiness	INTEGER		User's absolute trustworthiness	Value range is [0,100]	75
comm_status	ENUM(strStatus$^{\Delta}$)		User's current availability status		Idle
recent_device	ENUM(strDevice$^{\Delta}$)		User's currently-available or most-recently-used device		Cell3G
device_time	VARCHAR(19)		Most recent time when the user used the *recent_device*		2007-08-10. 14:59:46

$^{\Delta}$ strStatus: "Idle", "Busy".

$^{\Delta}$ strDevice: "Computer", "CellGSM", "Cell3G", "CellWiFi", "Phone", "IPVideoPlayer", "IPTV", "Unknown".

Table B-3. Table "personal_detail" records user personal details.

Field	Type	Feature	Explanation	Notes	Example
id	INTEGER	Primary key	Record index		1

user_id	VARCHAR(4)	Foreign key	User identification	Unique	AA11
location	ENUM(strLocation[Δ])	Foreign key	User's social domains		Home
device_type	ENUM(strDevice[Δ])		User's default available devices for each domain		CellGSM
device_code	VARCHAR(32)				+27-72-203-2817

Δ strLocation: "Home", "Office", "Club", "Public".

Δ strDevice: "Computer", "CellGSM", "Cell3G", "CellWiFi", "Phone", "IPVideoPlayer", "IPTV", "Unknown".

Table B-4. Table "social_relation" stores user social relations.

Field	Type	Feature	Explanation	Notes	Example
id	INTEGER	Primary key	Record index		1
user_id	VARCHAR(4)	Foreign key	User identification	Index to other user information	AA11
user_name	VARCHAR(45)		User name		Yang
related_user_id	VARCHAR(4)		ID of another user who has a relationship with the user		AA33
related_user_name	VARCHAR(45)		Name of the related user		Tao
relation_id	INTEGER	Foreign key	ID for the two users' relationship type	Index to table "relation_type"	10
trust_degree	INTEGER		The extent to which the related user trusts the principal user in general		100

Table B-5. Table "weekly_schedule" stores users' weekly social activities.

Field	Type	Feature	Explanation	Notes	Example
id	INTEGER	Primary key	Record index		1
user_id	VARCHAR(4)	Foreign key	User identification	Index to other user information	AA11
time_start	VARCHAR(8)		Start time of the hour in a day	A day splits into 24 hours with each as a timeslot	00:00:00
activity	Enum(strActivity[Δ])		User's activity during the one-hour period		OfficeIssue

* Table "week_schedule" factually comprises the table "week_schedule_workday" and the table "week_schedule_weekend".

Δ strActivity: "HomeIssue", "OfficeIssue", "LeisureIssue", "ClubIssue", "TrafficIssue", "OtherIssue".

The information provided in Table B-6 to Table B-9 provides a social-networking reference for the virtual-user system. Firstly, the information in Table B-6 and Table B-7 helps establish users' social relationships.

Table B-6. Table "relation_type" describes relationship types in the system.

Field	Type	Feature	Explanation	Notes
relation_id	INTEGER	Primary key	Record index	
type	ENUM(strRelation$^{\Delta}$)		Relationship type	
domain	ENUM(strDomain$^{\Delta}$)	Foreign key	Domain that the specific relationship belongs to	
demanding_ degree	INTEGER		How strongly the user requests the related user to process sessions	A range of [0, 1, 2, 3, …, 10]
counterpart_ relation_type	ENUM(strRelation$^{\Delta}$)		If the relation is from user A to user B, then its counterpair is that from user B to user A	Asymmetric to relationship *type*
counterpart_ relation_id	INTEGER		ID of the counterpair relationship type	

$^{\Delta}$ strRelation: "Husband", "Wife", "Parent", "Child", "ElderSibling", "YoungerSibling", "TheElder", "TheYounger", "Customer", "Supplier", "Partner", "Boss", "Colleague", "Underlying", "CloseFriend", "CommonFriend", "Instructor", "Learner", "Buddy", "Contacted", "Stranger".

$^{\Delta}$ strDomain: "Family", "Business", "Friend", "Club", "Other", "Unknown".

Table B-7. Enumerating the data of the table "relation_type".

relation_id	type	domain	priority	counterpair_relation_type	counterpair_relation_id
10	Husband	Family	9	Wife	11
11	Wife	Family	9	Husband	10
12	Parent	Family	8	Child	13
13	Child	Family	4	Parent	12
14	ElderSibling	Family	6	YoungerSibling	15
15	YoungerSibling	Family	5	ElderSibling	14
16	TheElder	Family	7	TheYounger	17
17	TheYounger	Family	3	TheElder	16
20	Customer	Business	10	Supplier	21
21	Supplier	Business	8	Customer	20
22	Partner	Business	7	Partner	22
23	Boss	Business	8	Underlying	25

24	Colleague	Business	5	Colleague	24
25	Underlying	Business	3	Boss	23
30	CloseFriend	Friend	7	CloseFriend	30
31	CommonFriend	Friend	4	CommonFriend	31
40	Instructor	Club	6	Learner	41
41	Learner	Club	4	Instructor	40
42	Buddy	Club	5	Buddy	42
50	Contacted	Other	2	Contacted	50
51	Stranger	Unknown	1	Stranger	51

Secondly, the information provided in Table B-8 and Table B-9 helps project users' social activities to their physical locations.

Table B-8. Table "activity_2_location" projects user activity to physical location.

Field	Type	Feature	Explanation	Notes
id	INTEGER	Primary key	Record index	
activity	ENUM(strActivity$^{\Delta}$)		User's activity during the time period	
location	ENUM(strLocation$^{\Delta}$)	Foreign key	User's physical position	

$^{\Delta}$ strActivity: "HomeIssue", "OfficeIssue", "LeisureIssue", "ClubIssue", "TrafficIssue", "OtherIssue".

$^{\Delta}$ strLocation: "Home", "Office", "Club", "Public".

Table B-9. Enumerating the data of the table "activity_2_location".

id	activity	location
1	HomeIssue	Home
2	OfficeIssue	Office
3	LeisureIssue	Home
4	LeisureIssue	Club
5	LeisureIssue	Public
6	ClubIssue	Club
7	TrafficIssue	Public
8	OtherIssue	Public

B.2 Application Database for the Application-Library Module

The information in Table B-10 to Table B-11 shows users' communication profiles.

Table B-10. Table "application_index" stores key information for all applications.

Field	Type	Feature	Explanation	Notes	Example
id		Primary key	Record index		5
application_id	VARCHAR(4)	Foreign key	Application identification	Unique	0031
application_name	VARCHAR(20)		Application name		VoiceCall _Phone
media_type	ENUM(strMedia$^{\Delta}$)		Type of physical media that transmits the application	Index to table "char_mediatype"	Audio
connecting_time	INTEGER		Period of setting up a connection for the application	In seconds	10
bandwidth	INTEGER	Foreign key	Required bandwidth for the application	Index to table "char_bandwidth"	4
cost_sender	INTEGER		Cost charged to sender	In units	1
cost_receiver	INTEGER		Cost charged to receiver	In units	1
device	ENUM(strDevice$^{\Delta}$)		User device that handles the application		Phone
service_type	ENUM(strService$^{\Delta}$)		Type of service that the application carries		VoiceCall

$^{\Delta}$ strMedia: "Data", "Audio", "Video", "AudioVideo", "DataAudioVideo".

$^{\Delta}$ strDevice: "Computer", "CellGSM", "Cell3G", "CellWiFi", "Phone", "IPVideoPlayer", "IPTV", "Unknown".

$^{\Delta}$ strService: "Email", "FileSharing", "VoiceCall", "VoiceMessage", "Messenger", "Video", "Teleconferencing".

Table B-11. Table "char_bandwidth" stores the characteristics of bandwidth.

Field	Type	Feature	Explanation	Notes	Example
level	INTEGER	Primary key	Level value shown in "application_index"		4
low_bound	INTEGER		Low bound of the bandwidth range		64
high_bound	INTEGER		High bound of the bandwidth range		128
unit	ENUM(strUnit$^{\Delta}$)				KHz

$^{\Delta}$ strUnit: "KHz", "MHz", "GHz".

Appendix C Database Information Relevant to Yang's Relationship Topology

The information provided in Table C-1 to Table C-5 all relates to Yang's social-relationship topology in Figure 6-1.

C.1 Selected Personal Details of Users

Table C-1. Relevant personal details of Yang.

User ID	User Name	Location	Device Type
AA11	Yang	Home	CellGSM, Computer
		Club	Phone, CellGSM

* Values are selected from the table "user_index" and the table "personal_detail".

Table C-2. Relevant personal details of Anthony.

User ID	User Name	Location	Device Type
AA22	Anthony	Club	Unknown
		Public	Cell3G

* Values are selected from the table "user_index" and the table "personal_detail".

C.2 Selected Social Relationships of Users

Table C-3. Relevant records of Yang's social relationships.

ID of Related User	Name of Related User	Trust Degree of Yang to the User	Relationship Type with Yang	Situated Domain	Demanding Degree of the User to Yang
AA22	Anthony	80	Boss	Business	8
AA33	Tao	100	Husband	Family	9
AA44	Deeya	75	Colleague	Business	5
AA55	Indika	73	Buddy	Club	5
AA66	National Security	100	Contacted	Other	2
AA77	YongJi	85	Father	Family	8

AA88	Samuel (Sam)	70	Learner	Club	4
AA99	Jackson	0	Stranger	Unknown	1
BB11	Terence	71	Instructor	Club	6
BB22	QiQi	78	Close Friend	Friend	7
BB33	JinJin	68	Common Friend	Friend	4
BB44	Louisa	98 (Louisa to Anthony)	Wife	Family	9 (Anthony to Louisa)

* Values are selected from the table "social_relation" and the table "relation_type".

Table C-4. Relevant records of Tao's social relationships.

ID of Related User	Name of Related User	Trust Degree of Tao to the User	Relationship Type with Tao	Situated Domain	Demanding Degree of the User to Tao
AA11	Yang	100	Wife	Family	9
AA44	Deeya	60	Common Friend	Friend	4
AA77	YongJi	70	TheElder	Family	7

* Values are selected from the table "social_relation" and the table "relation_type".

C.3 Selected Communication Schedules of Users

Table C-5. Relevant records of users' weekly schedules.

User ID	User Name	Start Time	End Time	Social Activity
AA11	Yang	19:00:00	19:59:59	ClubIssue
		20:00:00	20:59:59	HomeIssue
AA22	Anthony	18:00:00	18:59:59	ClubIssue
AA33	Tao	19:00:00	19:59:59	HomeIssue

* Values are selected from the table "user_index" and the table "week_schedule_workday".

Thank you for reading the thesis.