Reachability Management in a networked home

Johan Ensing[1,2], Mark H. Verberkt[1], and Leon J.M Rothkrantz[2]
1. Media Interaction group
Philips Research Laboratories
Prof. Holstlaan 4, 5656 AA Eindhoven
The Netherlands

2. Data and Knowledge group
Department of Information Technology and Systems
Delft University of Technology
P.O. Box 356, 2600 AJ Delft
The Netherlands

Keywords
Reachability Management, Context awareness, Ambient Intelligence

Abstract
This paper discusses a model for Reachability Management and the architecture of a Reachability Management System (RMS). On the one hand, a RMS must make sure that a person is not disturbed by communication, and on the other hand a RMS must take care that the person is aware of communication in time. The model for Reachability Management indicates how a person can control her reachability. The reachability of a person indicates what types of communication can be used to communicate with a person, and to what extent a person is willing to start a particular type of communication. The architecture defines the functionality and the operation of the components of a RMS. Furthermore, the architecture shows how the RMS can be integrated into the WWICE system. The WWICE system is a system that is developed at Philips Research and that (among other things) deals with issues concerning distributed computing in a home environment.

Introduction
The need for Reachability Management research results from the progress in (communication) technology that enables people to reach each other at all times, using all kinds of communication, no matter where people are. The drawback of this new technology is that you might receive communication when you don't want to be disturbed by communication, e.g. if your phone rings when you are sleeping. Reachability Management must give a person more control over her reachability. A system that manages the reachability of a person is called a Reachability Management System (RMS). On the one hand, the RMS must make sure that a person is not disturbed by communication, and on the other hand the RMS must take care that the person is aware of communication in time. In other words, the system must notify a person of incoming communication at the right time, in the right way (if it must notify a person at all).

An important aspect of reachability is the context information that is of importance to determine one's reachability (e.g. the activities and the location of people). Other important aspects are privacy (e.g. should a RMS inform a caller that the callee is having a holiday), and trust (will people trust a RMS to manage their communication). However, the latter two aspects are social consequences of using a RMS that are not in the scope of this paper.

The Reachability Management research is conducted as part of the Ambient Intelligence research at Philips Research [1], [2], [3]. Ambient Intelligence is characterized by ubiquity, transparency and intelligence. Ubiquity because the user is surrounded by a multitude of interconnected embedded systems, transparency because these systems are invisible and moved into the background of the user's surroundings, and intelligence because the system is able to recognize the inhabitants, adapts itself to them, learns from their behavior, and even shows emotion. Ambient Intelligence research investigates how systems can deal with ubiquity, transparency and intelligence. These systems are called ambient systems. The part of the Ambient Intelligence research that investigates how sensor technology can make interacting with computers easier is called Context Awareness. Context Awareness is an important research topic within the Ambient Intelligence research since it provides the information that an ambient system needs for more intelligent behavior.
The WWICE project is an example of a project at Philips Research that develops an ambient system [5], [6]. The WWICE project explores new applications in the electronic market that offer more entertainment, comfort, and flexibility in a networked home environment. It focuses on new application concepts, the corresponding interaction concepts for easy access and control, and the system architecture needed to support these concepts. In particular, the system architecture deals with issues concerning distributed computing. The WWICE system makes it very easy for people to communicate with each other, since the system can track (to some extent) the location of people

and devices, and it uses the devices near a person to start communication. However, people might not always want to be disturbed by communication. Therefore, there is a need for a RMS that enables people to control their communication.

Model for Reachability Management

The model that is presented in this paper is based upon eleven interviews and a literature survey [7], [8], [9], [10], [11], [4]. The RMS uses this model to determine whether someone is available for communication. The model consists of three components: the Priority manager, the Status manager, and the Threshold manager. The Priority manager determines the priority^r of the communication. The priority^r is the importance of the communication from the recipient's point of view[1]. The Status manager determines the status of the user. The status is the concept that enables a person to control her reachability. The status indicates to what extent a person is willing to start communication. Based upon the status, the Threshold manager determines the threshold of communication. The threshold is the minimum priority^r that a particular type of communication must have such that the recipient is available for that communication. If the priority^r is higher than the threshold, the recipient is available for that type of communication. Otherwise, she is not available. The following subsections explain the Priority manager, the Status manager, and the Threshold manager in more detail.

Priority Manager

An important aspect that determines whether someone is available for a particular type of communication is the importance of that communication from the recipient's point of view. This is called the priority^r. The Priority manager is the subsystem of the RMS that determines the priority^r. In this paper, the following classification for the priority^r is used: 'Normal', 'High', and 'Emergency'[2]. This classification is based upon the interviews that are conducted to determine this model for Reachability Management. Figure 1 illustrates the information that the Priority manager uses to determine the priority^r.

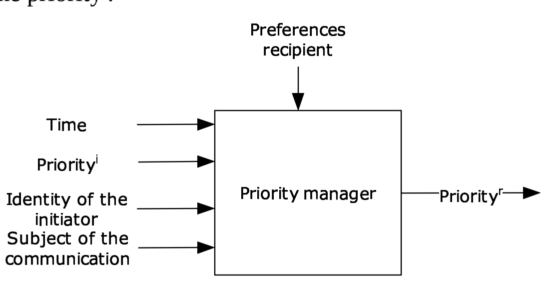

Figure 1 Information that determines the priority^r

[1] If person A indicates that she wants to start communication with person B, person A is called the initiator of the communication and person B is called the recipient.
[2] User tests must determine the best classification.

The user preferences of the recipient determine how the Priority manager combines the information that is shown in Figure 1. Per default the Priority manager sets the priority^r to 'Normal'. Next, the Priority manager adjusts the priority^r, based upon the information that is shown in Figure 1. The Priority manager uses knowledge rules to adjust the default priority^r:

Priority^i. The priority^i is the importance of the communication from the initiator's point of view. If the initiator has indicated a priority^i, this value is used as the default value for the priority^r.

Identity of the initiator. This information might pose upper or lower bounds to the priority^r, e.g. your girlfriend has at least the priority^r 'High' and your mother in law has at most the priority^r 'Normal'.

Subject of the communication. This information might also pose upper or lower bounds to the priority^r, e.g. the subject 'Meeting' has at least the priority^r 'High'.

Time. The higher the priority^r, the more effort the RMS puts into notifying the recipient. Therefore, a recipient might indicate that the Priority manager must increase the priority^r over time. This way the recipient might be aware of the communication in time. For example, she might indicate that the Priority manager must increase the priority^r of communication to 'High', one day after the communication arrives at the RMS. Obviously, the recipient will only indicate this preference for non real-time communication. If the communication is real-time the initiator will not wait a day for the response of the recipient. Real-time communication is communication whereby two persons have direct interaction. In other words, the interval of time between one person sending a message (asking a question) and the other person replying with a message (answering the question) is negligible. Some examples of real-time communication are telephone communication and video communication. Non real-time communication is communication that is not real-time, for example email and SMS

Of course, the Priority manager must contain a mechanism that detects inconsistencies in the preferences of the user, such that it can inform the user in advance whether her preferences cause conflicts. When conflicts occur due to other reasons, e.g. your girlfriend (lower bound "High") sends a message with subject "WIN 1 MILLION DOLLARS" (upper bound "Normal"), the Priority manager assigns the highest priority^r to the communication to ensure that the recipient does not miss important communication.

Status manager

The status is the concept that enables the user of the RMS to control her reachability. For example, if the status of a user is 'Busy' this might mean that she is not available for real-time communication. The Status manager is the subsystem of the RMS that determines the status. User tests that are performed at Microsoft research conclude that users do not want to use many statuses [4]. In this paper, the statuses ‘Available’, ‘Busy’, ‘Working’, and ‘Not available’ are used to explain how the concept ‘Status’ works. Figure 2 illustrates the information that influences the activation of statuses.

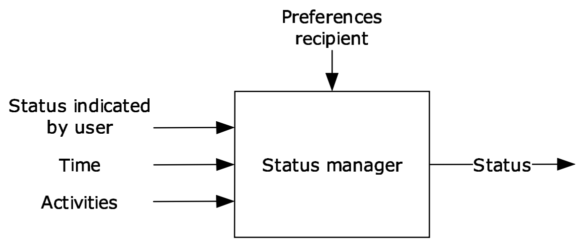

Figure 2 Information that might activate statuses

The preferences of the recipient determine how the Status manager combines the information that is shown in Figure 2. Per default, the Status manager sets the status to ‘Available’ (which means that the recipient is available for all communication). The Status manager adjusts the status, based upon the information that is shown in Figure 2:

Status indicated by user. A user can indicate her status directly, using the UI of the RMS. For example, a user might indicate that for the next hour, her status is ‘Not available’.

Time. The user might indicate in advance that she has a particular status during some periods, e.g. her status is always ‘Not available’ between 0:00h and 7:00h.

Activities of the user. The user might indicate that she has a particular status during particular activities, e.g. her status is ‘Busy’ when she is in involved in the activity ‘Watching the news’, and her status is ‘Not available’ when her activity is ‘Sleeping’.

Of course, the Status manager must contain a mechanism that detects inconsistencies in the preferences of the user, such that it can inform the user when her preferences cause conflicts. The Threshold manager needs *one* status to determine the threshold. During a day, conflicts might occur because multiple statuses are activated at the same time. For example, the time is 0:30 (this means that the user’s status is ‘Not available’) and the user indicates, using the User Interface of the RMS, that her status is ‘Available’. In order to solve conflicts, the Status manager determines the ‘user status’. This is the status that the Threshold manager uses to determine the threshold. From the interviews results that the Status manager should determine the user status as follows: per default, the user status is ‘Available’. If the user indicates a status directly, then this is the user status. Otherwise, the Status manager uses the status that has the strongest impact on the user’s reachability. In other words, it uses the status that makes the user the least reachable. This implies that the user must indicate for each status the impact of that status on her reachability. For example, the user could order the statuses as follows: [1] Not available; [2] Working; [3] Busy; [4] Available.
This means that the status ‘Working’ is the user status, when both the status ‘Working’ and the status ‘Busy’ are activated, and the user did not indicate a status directly. The Status manager uses the status that has the strongest impact on one’s reachability since the user activates statuses to indicate that she does *not* want to be disturbed. Thus the status that makes sure that she is not disturbed, must become the user status. This way it is easy for the user to understand how the Status manager determines the user status.

Threshold manager
The threshold is the minimum priority^r that communication must have such that the recipient is available for the communication. The Threshold manager determines the threshold and it determines, based upon the threshold and the priority^r, whether a recipient is available for a particular type of communication. If the priority^r is higher than the threshold, the recipient is available. Otherwise, she is not available. Figure 3 illustrates the information that influences this decision process.

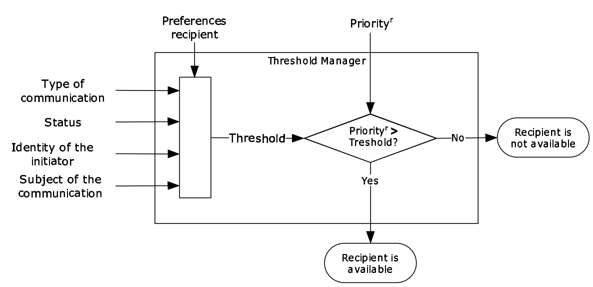

Figure 3 Information that determines the threshold

From interviews results that the default threshold value mainly depends on the type of communication and the status. For example, when the user status is ‘Working’, the threshold for all real-time communication might be ‘High’. The Threshold manager adjusts the default threshold based upon preferences with respect to the status of the recipient in combination with the identity of the initiator and/or the subject of the communication. For example, a user might indicate that when her status is ‘Working’, the threshold for all communication from colleagues is ‘Normal’ (instead of ‘High’). When conflicts occur, the Threshold manager uses the default threshold.
The Threshold manager determines whether a recipient is (still) available for communication, every time it determines a new threshold, or when the Priority manager determines a new priority^r. The Threshold manager determines a new threshold, every time the Status manager determines a new status.

Architecture of RMS

An architecture has been developed that demonstrates the use of a RMS. This architecture shows which components are necessary to obtains the functionality of a RMS. A RMS provides the following functionality:

Reachability Management. When the RMS must deal with incoming communication, it determines whether it is possible to start that communication (e.g. whether enough bandwidth is available to start video communication), and whether the recipient is available for that type of communication (based upon the model for Reachability Management).

Notification. When the RMS determines that it is possible to start communication, and the recipient is available for communication, it notifies the recipient.

Communication rejection. When the RMS determines that it is not possible to start communication, or the recipient is not available for communication, it rejects the communication. It might forward the communication, it might start monitoring (wait until it is possible to start the communication and the recipient is available), and it might start interaction with the initiator (e.g. propose to record a voice-mail message). Per default, the RMS starts interaction with the initiator when the communication is real-time, and it starts monitoring when the communication is *not* real-time.
Start communication. When the initiator and the recipient agree to start a particular type of communication, the RMS starts the communication, e.g. it set-ups video communication between the device that the recipient is using and the device that the initiator is using. It uses the available home system to start the communication, e.g. the WWICE system.

Learning. The RMS tries to extract new preferences based upon the behavior of a person.

Figure 4 shows the components of the RMS.

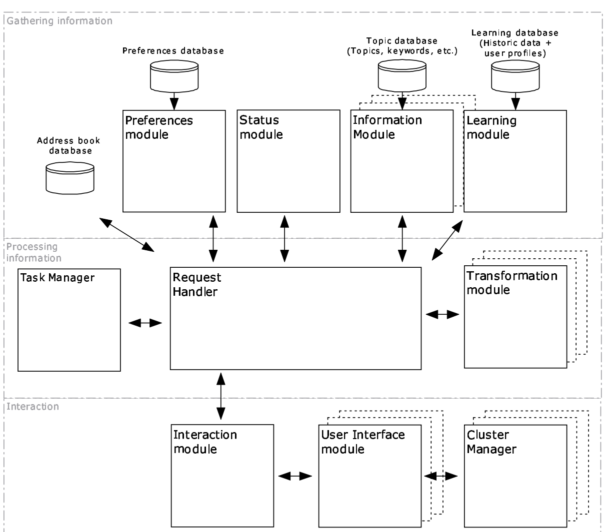

Figure 4 Components of RMS

Figure 4 shows only the most important interaction between these components. The actual interaction is not in the scope of this paper.

Task Manager

This component distributes tasks to Request Handlers. For example, when a communication request arrives, e.g. someone wants to start a telephone call, it starts a Request Handler to deal with this communication request. Furthermore, the Task manager takes care of communication between RMS-es.

Request Handler

The Request Handler deals with all tasks of a RMS. The main tasks are ‘start communication’ and ‘deal with incoming communication’. If it must start non real-time communication, it asks the WWICE system to start the corresponding communication application (e.g. an email application if the initiator wants to send an email message). If it must start real-time communication, it sends a communication request to the RMS of the recipient. If it must deal with an incoming communication request, it determines whether it should notify the recipient (it notifies the recipient if it is possible to start the communication and if the recipient is available for the communication). If it determines that it should notify the recipient, it instructs the Interaction module to take care of this. If it determines that it should not notify the recipient, it takes care of the communication rejection. This means that it might start an Interaction module to start interaction with the initiator, it might forward the communication, or it might start monitoring. Furthermore, the Request Handler monitors the progress of the notification and it undertakes actions when the recipient does not notice the communication in time. If the recipient and initiator agree to start a particular type of communication, the Request Handler asks the Application Manager to start the corresponding communication application. The Application Manager is an entity of the WWICE system that starts, given a URN, the corresponding application at the correct device. This means for example, that if the Request Handler provides the URN that stands for the telephone number of a person, the WWICE system creates a telephone connection and starts a telephone application.

Interaction module

The interaction module takes care of the interaction between a RMS and a person. It determines the content of the interaction. More specific, it determines an interaction scheme that consists of the following parts:

- The information that must be transferred, e.g. the Interaction module can inform an initiator that the recipient is not available for communication.
- The options that a person can choose from, e.g. an initiator can choose to record a voicemail message.

If the Interaction module must start interaction with a person that is using a remote RMS (e.g. an initiator), it asks the Request Handler to send the interaction scheme to the remote RMS. If it must start interaction with a person who's location is in the home where the RMS is installed, it starts a User Interface module and sends the interaction scheme to the User Interface module.

User Interface module
A User Interface module takes care of the User Interface (UI) of the RMS. It determines which clusters it must use for the User Interface. Within the WWICE system, devices are grouped in clusters based upon location. An application can create one UI per cluster of devices. For example, there might be a cluster that contains a screen and a PDA. A Television application can use the screen to display content (a television show), and it can use the PDA to display the UI for the control of the application.
The User Interface application determines a UI description based upon the interaction scheme that it received from the Interaction module. This UI description contains the 'UI widgets' and the properties of these UI widgets. A UI widget indicates a particular type of UI building block, e.g. a button. The User Interface module creates one UI description per cluster, since every cluster might have different properties (e.g. every cluster might have different UI capabilities available). It sends the UI descriptions to the appropriate Cluster Managers.

Cluster Manager
The RMS needs a Cluster manager to deal with the situation that multiple User Interface modules must create a UI in the same cluster of devices, e.g. one User Interface module must create a UI for the notification of new email and another User Interface module must create a UI for the notification of an incoming telephone call. The Cluster manager combines the UI descriptions of different User Interface modules in order to create one WWICE-specific UI description for a cluster. Furthermore, the Cluster manager determines the specific properties of a cluster (e.g. the available UI capabilities, whether someone is using the cluster, and the activities that occur nearby the cluster). The Cluster manager sends its' UI description to the appropriate UI Managers. A UI Manager is a WWICE entity that combines UI descriptions from different applications in order to create one UI for a particular device.

Status module
The Status module determines the status of the recipient. The status is the concept that allows a user of the RMS to control her reachability. The Status module implements the 'Status manager' that is explained above.

Preferences module
The Preferences module takes care of personalization. The Preferences module stores all preferences of the users of the RMS in the Preferences database. The components of the RMS can request these preferences. An example of a preference is a preference that indicates when someone wants to have a particular status, e.g. a user wants to have the status 'Not available' between 0:00h and 7:00h.

Information module
An Information module provides a service that extracts information from available data. For example, there might be an Information module that is able to extract the subject of a communication request from a text. The Request Handler might use an Information module, for example to determine the subject of an email message.

Transformation module
A Transformation module provides a service that can transform one type of communication into another. For example, there might be a Transformation module that can transform a voicemail message into an e-mail message. The Request Handler might use a Transformation module, if it determines that a recipient is not aware of communication in time and it determines that it should forward the communication to another communication channel. For example, if it wants to forward an e-mail message to a mobile phone, it needs a Transformation module to transform the e-mail message into a SMS message.

Learning module
The learning module stores all kinds of data in a Learning database (e.g. the response of a user when a particular initiator is calling), and tries to extract new user preferences.

Conclusions
This paper presents a typical example of Ambient Intelligence research. Ambient Intelligence research and related research areas such as Context Awareness [9] and Intelligent environments [10], are currently hot research topics. The 'ambient' part of this research project is that this project explores ways to use context information (e.g. the activities that a person is involved in), to give a person more control over her reachability. Furthermore, this project presents the architecture of a Reachability Management System (RMS) that can be incorporated into an ambient system. From the work that is presented in this paper, a number of conclusions can be drawn. These conclusions are presented below.

This paper presents a realistic model that demonstrates how a user can control her reachability. Some important issues that played a part during the development of the model for Reachability Management, are that average consumers must be able to understand the model, and that the model must deal with contradictory information. For example, user preferences might indicate that the priority of communication must be both high (e.g. because the initiator is important) and low (e.g. because the subject of the communication is not interesting). The model that is presented in this paper is easy to understand, and it indicates how the RMS should deal with contradictory information. Further research should refine the model based upon user tests.

This paper identifies the basic functionality that a RMS must have with respect to Reachability Management in a home environment. It is possible to come up with more functionality, especially if one increases the scope to the office environment. However, the functionality that is presented in this paper satisfies to demonstrate the use of a RMS.

This paper presents the architecture of a RMS that introduces the components that are necessary to obtain the basic functionality of a RMS. This architecture defines the functionality of the components and shows how the components interact with each other, and how they interact with the entities of the WWICE system. This architecture satisfies to demonstrate the operation of a RMS and to demonstrate how the RMS can be implemented in the WWICE system.

The architecture that is presented in this paper can easily be implemented in the WWICE system. Moreover, a substantial part of the architecture is not WWICE-specific. This means that the architecture can also be used to implement the RMS in another (home) system.

Recommendations

The work that is presented in this paper can be used as a basis for further research. Further research could investigate the following issues:

Functionality of RMS and model for Reachability Management

In order to test the use of a RMS and the model for Reachability Management, one should implement the architecture of the RMS and conduct user tests that investigate (among other things) the following properties of the RMS:

- The benefit of the functionality of a RMS. In other words, does the RMS provide functionality that a user finds valuable?
- The clarity of the model for Reachability Management. In other words, is it clear to a user *how* the RMS determines whether she is available for communication?
- The benefit of the model for Reachability Management. In other words, does the model cover all preferences that a user wants to indicate with respect to her availability for communication, and does a user find the model easy to use?
- Privacy. What information should a RMS provide to people (e.g. should a RMS inform a caller that the callee is having a holiday), and what information is a RMS allowed to store (e.g. is the RMS allowed to store the number of times that a person is having a telephone conversation with someone)?
- Trust. How can the RMS make sure that people trust the RMS to manage their communication?

User Interface

From the interviews results that people are easily afraid that a RMS will block communication that they *do* want to receive. In important issue that increases the trust of people is the User Interface (UI) of the RMS. Further research should investigate how the UI can show in an easily understandable way, how the RMS manages the communication of a user. Furthermore, it must be very easy for users to indicate preferences and to indicate their current status via the UI of the RMS.

Activity Monitor

The architecture of the RMS takes the activities of the user into account. It retrieves these activities from an Activity Monitor that is assumed to be part of the WWICE system. However, the Activity system is not implemented in the WWICE system. Further research must determine how the Activity Monitor should work exactly. The main research questions are the abstraction level of an activity (e.g. whether the Activity Monitor most provide the information 'user is sleeping', or 'user is not moving'), and how the Activity Monitor can detect activities (e.g. what types of information are necessary to detect an activity, and how can the Activity Monitor combine this information to determine that an activity occurs).

Exact operation of the components of the RMS

This paper presents a thought-out architecture of a RMS. This architecture demonstrates the operation of a RMS and demonstrates how the RMS can be implemented in the WWICE system. However, further research is required to determine how the components must work in detail. For example, further research must investigate how the Learning module can determine new preferences (e.g. using neural networks or data mining techniques).

References

[1] Aarts E.H.L. and Harwig H.A., Ambient intelligence: a new user experience. *Cebit 2000*, 2000.

[2] Aarts E.H.L., Ambient Intelligence: calming, enriching and empowering our lives. *Philips Research Password*, 8, 2001, http://www.research.philips.com/InformationCenter/Global/.

[3] Aarts E.H.L., Converting dreams into real experience. *World News*, 2001, http://www.research.philips.com/InformationCenter/Global/.

[4] Rannenberg K., "Multilateral Security A Concept and Examples for Balanced Security". ACM Press, Cork, Ireland, 2000, http://csrc.nist.gov/nissc/2000/proceedings/papers/202ra.pdf.

[5] Eggenhuizen H.H., *WWICE: Window on the World of Information, Communication and Entertainment*, Philips Research Password, 2000.

[6] Simons D. and Reuzel J., *WWICE Software architecture - only available within Philips Research*, CRE 1999, 1999.

[7] Johannes S. and Marti W., *Active Messenger: Email Filtering and Mobile Delivery*. Thesis for the degree of Massachusetts Institute of Technology, 1999, http://citeseer.nj.nec.com/211552.html.

[8] Görg C., Farjami P., Bell F., Hagen L., Magedanz T., Vodslon M., Weckerle C., Vortisch W., Mauersberger J., Jaya S., Chandrasekaran V., Loryman M., Buckle P., Major B., Bretzke S., Hartmann J., Song W., Evensen R., Lülsdorf A., Kleier S., and Timphus F., *CAMELEON – Communication Agents for Mobility Enhancements in a Logical Environment of Open Networks*, 1998, http://www.comnets.rwth-aachen.de/project/cameleon/cameleon.html.

[9] Hartmann J., Gorg C., and Farjami P., "Agent Technology for the UMTS VHE Concept". 1998, www.jens-hartmann.de/papers/vhe.pdf.

[10] Raman B., Katz R.H. , and Joseph A.D., "Universal Inbox: Providing Extensible Personal Mobility and Service Mobility in an Integrated Communication Network". Workshop on Mobile Computing Systems and Applications (WMSCA'00), 2000, http://www.cs.berkeley.edu/~bhaskar/research.html.

[11] Reichenbach M., Damker H., Federrath H., and Rannenberg K., "Individual Management of Personal Reachability in Mobile Communication". 13th International Conference on Information Security (SEC `97), 1997, http://citeseer.nj.nec.com/479072.html .