A Need-awaring Multi-agent Approach to Nomadic Community Computing for Ad Hoc Need Identification and Group Formation

Keunho Choi
Graduate School of International Management
(kino4u@khu.ac.kr)

Ohbyung Kwon
Graduate School of International Management
(obkwon@khu.ac.kr)

Recently, community computing has been proposed for group formation and group decision-making. However, legacy community computing systems do not support group need identification for ad hoc group formation, which would be one of key features of ubiquitous decision support systems and services. Hence, this paper aims to provide a multi-agent based methodology to enable nomadic community computing which supports ad hoc need identification and group formation. Focusing on supporting group decision-making of relatively small sized multiple individual in a community, the methodology copes with the following three characteristics: (1) ad hoc group formation, (2) context-aware group need identification, and (3) using mobile devices working in- and out-doors. NAMA-US, an RFID-based prototype system, has been developed to show the feasibility of the idea proposed in this paper.

Keywords : Community computing, group need identification, ubiquitous computing, ad hoc group formation, multi-agent system

Received: November 2005 Accepted: June 2006 Corresponding Author: Keunho Choi

1. Introduction

Ubiquitous computing technology becomes promising to realize next generation services. While walking around with mobile devices, nomadic users tend to make decisions for problem solving by accessing information, communicating with others, and thinking for their education, business, research, and general entertainment. Frequently, the decisions can be made collectively. For example, a user who is on shopping and about to purchase a product may want to get advised by her/his friends or persons who already have experiences to use the product. An employee who is going to negotiate with her/his client may want to collect relevant information from others about the client on her/his way to the client's office. A doctorial candidate attending a conference may want to find a right person to appear in the job market. A jobless person wants to find an appropriate employer as soon as possible when she/he is walking around downtown. In short, a nomadic person needs persons who have complementary needs and hence will collaborate or coordinate with her/him any time, anywhere, by forming a group no matter what the group is static or dynamic.

Size of Group	Multiple Individuals	Single group	Multiple groups
Large	Local area decision net	Decision room	Electronic video conferencing
Small	Computer conference	Decision conferencing	Teleconferencing / Broadcasting

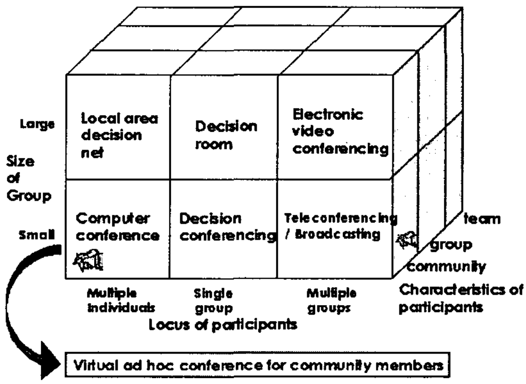
Locus of participants

[Fig. 1] Classification of group support systems

Meanwhile, group decision support system (GDSS) or collaboration support systems have aimed to support group or team to make decisions. An excellent review of group decision support systems has already been developed since the 1980's (Chun, and Park, 1998; DeSantis and Gallupe, 1987; Kraemer and King, 1988). Group support systems are generally classified with two dimensions: size of group and locus of participants, as shown in [Fig. 1].

To apply the classification of GDSSs shown in [Fig. 1] to ubiquitous computing environment settings, one more dimension is needed: characteristics of participants. The characteristics of participants consist of team, group, and community. Team and group are similar to each other in a sense that the participants have a same goal or interests. Groups are assemblages of persons gathered or located together. Teams are groups organized to work together. Comparing with group or team, community is characterized by its purpose and interaction in an anonymous manner. They do not necessarily know each other a priori. Community may be defined as a group of people living in a particular local area.

In this paper, we focus on decision support activity for relatively small sized multiple individual in a community. The decision support activity consists of negotiation, idea generation, voting, knowledge share, auction, and reverse auction. Comparing with typical group decision support systems, we aim to enable group to make decisions regardless of time and place. To do so, ad hoc and context-aware group formation should be supported while the community members carry mobile devices in- and out-doors. Moreover, the group members are assumed to be anonymous to each other. [Fig. 2] shows our focusing area among the group multiple individuals, virtual ad hoc conference is basically based on group formation from community members, rather than group or team members who are well acquainted with each other and share common goals to be together.



[Fig. 2] Locus of virtual ad hoc conference

Ad hoc group formation for any services in a community is expected to increase the problem solving abilities. To realize ad hoc group formation, how to correctly identify the nomadic group's ad hoc needs in an automated and seamless manner. However, research of ad hoc group need identification is still very few.

Hence, the purpose of this paper is to propose a community computing methodology to identify the ad hoc needs of the nomadic community. Moreover, when considering CBB (Consumer Buying Behavior) model, need identification phase, which has not been supported by legacy agent technology, is stressed in this paper because group formation is substantially related to theidentification of group formation needs. Web service based multi-agent architecture is adopted for the need identification mechanism. A prototype system, NAMA-US, is being implemented to show the feasibility of the methodology proposed in this paper.

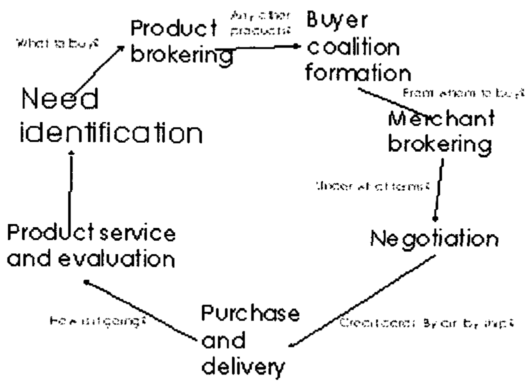
The remaining part of this paper is organized as follows. Section 2, we review the research background on need identification and community computing. In section 3, we suggest our framework and methodology for ad hoc group formation. Then, to show the feasibility of the idea proposed in this paper, a prototype system, NAMA-US, is illustrated in section 4. Finally, in section 5, we conclude with the research schedule and implications.

2. Literature Research

2.1 Need Identification

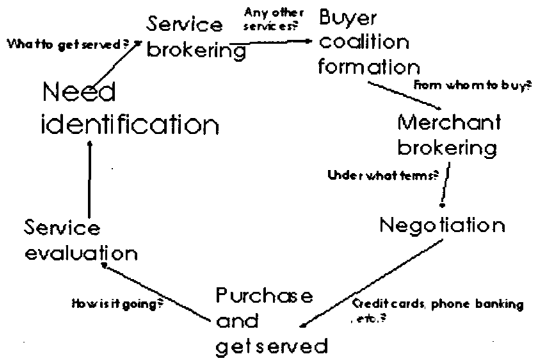
"Need" is defined as a sort of internal state to do something particular. Need is to a gap between what "is" and what "should be" (Witkin and Altschuld, 1995). Reviere mentioned that need as a gap between actual and ideal identified as community value (Reviere, et al., 1996). On the other hand, McKillip interpreted need as wants or as a demand (McKillip, 1987). In short, need could be explained by a current condition, wants and its difference.

Need awareness originates from recognizing unfulfilled needs. CBB model is a well-known customer model which includes need identification as a main component. The CBB model proposed by He et al. is shown as [Fig. 3] (He, et al., 2004).

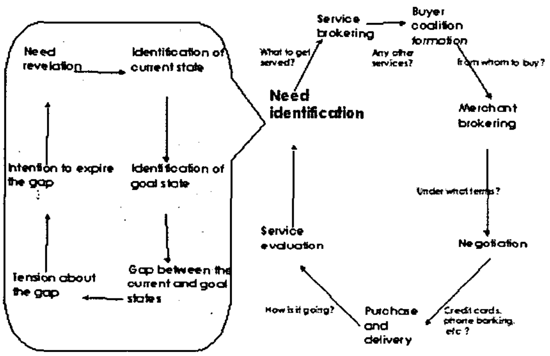


[Fig. 3] Basic CBB model

When focusing on buying behavior of services, the original CBB model may be amended as [Fig. 4]. Need identification is the first step of B2C procedure that enables online customers to buy goods from the e-marketplace (Blake, 2002). Since need identification is related to the customer's purchase behavior, it focuses more on predicting the future and what will be done, rather than what has been done. This results in the difficulties of deriving user need from one's past behavior. More detailed description of need identification of service-oriented CBB model is shown as [Fig. 5].



[Fig. 4] Service-oriented CBB for ubiquitous services



[Fig. 5] Need identification and service-oriented CBB model

2.2 Community Computing

Community computing began with the development of the timesharing computer in the early 1960s. The timesharing computer systems created a community by making use of network and corresponding information systems such as teleconferencing, groupware and electronic mail.

To understand community computing, the terminology community needs to be identified. The term "community" has been defined in the literature in different ways; for example Elisabeth Mynatt (Mynatt, et al., 1997)saw community as a social grouping which exhibit in varying degrees: shared spatial relations, social conventions, a sense of membership and boundaries, and an ongoing rhythm of social interaction.

The main differences between community and group are the level of interaction between the members and the existence of shared goals and artifacts. It should be noted that there is no clear separation between these group types. Seamless transitions occur between them and groups and teams can exist inside communities.

The term community computing can be defined as the methodologies and tools for creating, maintaining, and evolving social interaction in communities (Ishida, 1997). Community computing is intended to support more diverse and amorphous groups of people. Community computing typically supports the process of organizing people who are willing to share some mutual understanding and experiences. In other words, compared with groupware, community computing focuses on an earlier stage of collaboration: group formation from a wide variety of people. Community computing supports different functions for encouraging social interaction.

The functionalities of community computing are (1) finding someone to collaborate with, (2) making contact with the selected people, (3) building a common understanding, (4) collaborating with others in the same community, (5) executing individual work, and (6) communicating between co-workers in order to coordinate activities and work plans.

The main usefulness of communities lays in its being a starting point for identifying a set of people one could interact with, e.g. to find some help for solving problems or to share experiences. The basic rationale for community computing, as for any computer- supported co-operative work, is that people who want to communicate and collaborate are not always in the same place at the same time (Carroll, 2001).

The Blacksburg Electronic Village (BEV; http:// www.bev.net) have documented how various groups in the community organize their own activity and provide leadership to others in the community through the community network (Carroll and Rosson, 1996; Farrington and Pine, 1997). The BEV provides access to a huge volume of information and services; including health information from local doctors, local bus schedules, projects by school children, the rugby football club schedule, discussion of regional power line and highway proposals, information on area museums and a string quartet, and access to the Internet. Such a service enabled the residents of Blacksburg, Virginia to build up efficiently subcommunities. However, despite its idea and actual contributions, BEV so far does not assume ubiquitous computing environment settings.

Digital City Kyoto is to make it real by establishing a strong connection to physical Kyoto, by providing an information center for everyday life for actual urban communities (Ishida, 2002).

Communityware is intended to support more diverse and amorphous groups of people. Communityware typically supports the process of organizing people who are willing to reach some mutual understanding.

Familyware is a communication tool that provides status information about remotely located family members using peripheral displays and devices (Go and Carroll, 2003 Go, et al., 2000). By using the Internet and everyday artifacts, parents can take care of their own children with less effort, and children can communicate with their family in an easier way.

The MOOsburg project is focused on developing a community the Internet resource for Blacksburg. The purpose of the project is to create a community based on-line resource modeled on the town of Blacksburg in southwest VA, USA (Carroll, et al., 2001a; Carroll, et al., 2001b).

Despite several trials, full implementations of the community computing mechanism are still very few. Ad hoc group formation, which is a main purpose of community computing, is not so far based on need-awareness. To intelligently and seamlessly form a group among a community, the community members' needs should be identified in an automated and autonomous manner.

2.3 Matchmaker

<Table 1> shows the representative match-making methods which have been proposed in LNCS from 1998 to 2005 classified with matchmaking knowledge learning, matchmaking modes, and triggering methods. Among them, a few methods were related to community computing: Yenta and White Rabbit.

Yenta is a matchmaking system to find people having similar interests and encouraging communication between them (Foner, 1998). To represent the user's similar interests, decentralized agents of the system group into categories by using the Internet web mail and files. One of the disadvantages of a centralized server is an accidental breakdown which can have severe consequences for reliability and availability of a system at all times. "*However, White Rabbit improved this architecture by adding a mediator agent, which facilitates the communication between the agents, and provides a single rally-point for clustering* (Thibodeau, et al., 2000)."

Unlike Yenta and White Rabbit, users of NAMA-US are assumed to be nomadic with portable mobile devices such as PDA. The NAMA-US agent acquires context data, users' profiles and service ontology. In addition, we propose a system not only to make a match actively for the request of users but also to provide the users with necessary on-line services proactively.

In accordance with security issues, we employ three security levels in terms of social relationship: private, protected and public. At private level, the user can prohibit all privacy concerning data to be shared by any other agents and persons. Protected level means that the user allows her/his privacy concerning data to be transferred to those who are well-known and trustworthy. In general, they could be the user's family, mates or colleagues. The range of protection is definitely declared by the user with the aid of user agent. At public level, the personal data are widely open to any persons and agents so that any proactive services can be reached to the user automatically. To do this way, we can make to relieve security issues in a way to follow pre-defined steps approximately using context data. Moreover, the system provides matchmaking services in accordance with social relationship rather than pure anonymity.

<Table 1> Classification of matchmaking

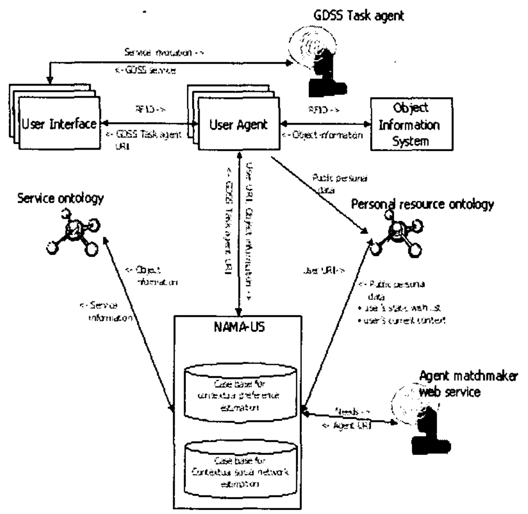
Matchmaking Knowledge Learning	Clustering	Foner, L. N. (1998), Thibodeau, M., et al. (2000), Lee, K., et al. (2004)
	Constraint-/rule- based	Freuder, E. C. and Wallace, R. J. (1998), Lee, K., et al. (2004)
	Heuristics	Tesch, T. and Fankhauser, P. (1999)
	Filtering	Eiter, T., et al. (2001), Sumi, Y. and Mase, K. (2001)
	Neural Network	Lee, K., et al. (2004)
Matchmaking Mode	Human-to-human	Foner, L. N. (1998), Freuder, E. C. and Wallace, R. J. (1998), Thibodeau, M., et al. (2000), Sumi, Y. and Mase, K. (2001), Hamasaki, M. and Takeda, H. (2003), Vivacqua, A., et al. (2003), Lee, K., et al. (2004), Hishiyama, R. and Ishida, T. (2005), Calì, A. (2005)
	Human-to-task agent	Van Dyke, N. W., et al. (1999), Lu, H. (2003), Satoh, I. (2003), Shen, R., et al. (2003)
	Task agent-to-task agent	Tesch, T. and Fankhauser, P. (1999), Wombacher, A. and Mahleko, B. (2002), Lu, H. (2003), Lee, K., et al. (2004)
Triggering Methods	Passive	Van Dyke, N. W., et al. (1999), Eiter, T., et al. (2001), Sumi, Y. and Mase, K. (2001), Wombacher, A. and Mahleko, B. (2002), Lu, H. (2003), Lee, K., et al. (2004)
	Active	Foner, L. N. (1998), Freuder, E. C. and Wallace, R. J. (1998), Tesch, T. and Fankhauser, P. (1999), Thibodeau, M., et al. (2000), Hamasaki, M. and Takeda, H. (2003), Satoh, I. (2003), Vivacqua, A., et al. (2003), Shen, R., et al. (2003)

In addition, the matchmaking methods can be categorized in terms of service continuity: permanent, temporal and conditional.

3. Multi-Agent Community Computing For Ad Hoc Group Formation

3.1 Framework

The proposed system framework of multi-agent system for ad hoc group formation and decision support is shown as [Fig. 6] User agent on behalf of the user gets RFID data from the user's currently using device with RFID reader attached. The user agent then requests object information of the entity, which contains RFID to Object Information System (OIS) by sending RFID data. The object information is regarded as context data: name of place, name of service and service URI, if any. For example, an RFID may indicate restaurants, dormitory, or train.



[Fig. 6] System framework

If the place has its own service ontology, then the URI of the service ontology is delivered together as object information to NAMA-US, a group formation agent, when the user agrees to get served by NAMA-US through her/his mobile device. NAMA-US is running all the time so that group formation service request is available in anad hoc manner. NAMA-US aims to identify group need of the users who are approximate with each other with the use of user URI and object information. To do so, the user's wish list is imported from the personal resource ontology. Moreover, to estimate remaining implicit needs which are not explicitly declared in the wish list, two case bases are used: case base for estimating contextual preference and contextual social network. A case based reasoning is performed to identify estimated needs with the corresponding case base, which contains personal profile and context data as conditions and revealed needs as results. The contextual need is estimated by the following equation:

$$Contextual\_need_{i,j} = f(s_1, s_2, \cdots, s_n | c_1, c_2, \cdots, c_m) \quad (1)$$

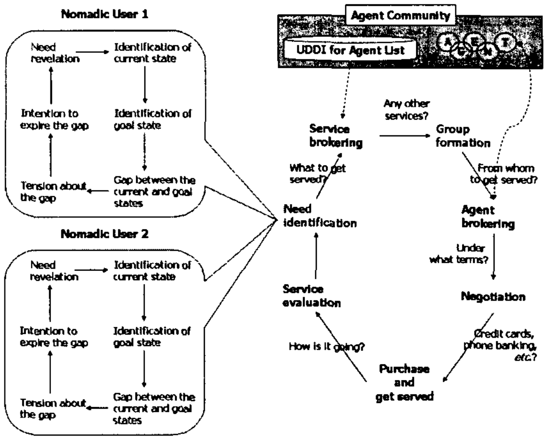
,where i and j indicates ith user and jth need, respectively. Ck is the kth contextual feature; Sl is the lthstatic feature. Static feature is characterized by the user's profile which comes from personal resource ontology. Contextual feature is represented as social context and context model.

Case base for contextual social context is adopted to selectively identify group needs that can be revealed a specific type of social context. For example, if the profile of the users in vicinity is their parents or their children, they may not want to buy and sell books or any other items.If the users are professor and students, they will not share information about anticipated questions and answers that will be appear in the upcoming exam. As a result, case based reasoning of contextual social context will narrow down the group needs which are derived from the estimated needs that are recommended by the contextual preference estimation phase and also from the declared needs of wish list. The group needs are intersection of needs of the users in the vicinity based on the RFID-attached place.

With the identified group needs, NAMA-US then invokes agent matchmaker web service, which is an agent-oriented web service. Agent-oriented web service is an agent that communicates with other agents or computer resources over web service protocol, SOAP. Hence, agent-oriented web service makes use of web service functionalities such as easy service deployment and discovery and standardized communication method.

The agent URI is returned to the NAMA-US, so that NAMA-US may inform available GDSS task agent to each user agent. The user agent then generates a dynamic web document and transmits it to the user interface. The user can invoke the selected GDSS task agent to get group service served.

The core of the system framework follows the service-oriented CBB model with need identification model as shown in [Fig. 7]. Each step of need identification model generates static and contextual content thatmight be useful for automated identification of group member's explicit and/or implicit needs as listed in <Table 2>.



[Fig. 7] Group needs identification and service-oriented CBB model for group works

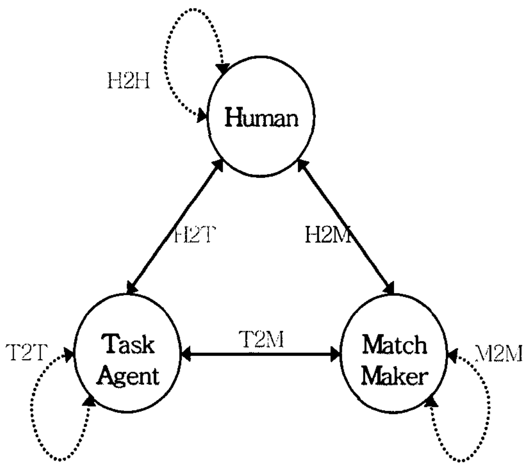
<Table 2> Steps in identifying group member's needs

Steps	Static content	Contextual content
Identification of current state	User profile	Current location Current time Current activity Current identity
Identification of goal state	Wish list To get served list	
Gap between the current and goal states		
Tension about the gap	Perceived sensitivity of contextual pressure	Contextual pressure Due duration of get served
Intention to expire the gap	Eagerness to get served	Social context Availability of services
Need revelation		Users' commitment to get served

Static content is imported from personal ontology or manual input. Contextual content is imported from sensors such as RFID and partially from E-Wallet.

3.2 Matchmaking Mode

As shown in <Table 1>, the matchmaking mode so far considered has been limited to human-to-human, human-to-task agent, and task agent-to-task agent. In this paper, a full-fledged matchmaking model is proposed as follows [Fig. 8]. The proposed architecture for classification of matchmaking mode is shown as [Fig. 9].



[Fig. 8] Matchmaking modes

Human-to-Human (H2H)

Human-to-human (H2H) is a basic mode to provide a matchmaking service. When a matchmaker receives a request from a user, it provides information from an appropriate service provider to the user. The matchmaker also provides the service provider such as task agent with the user requests and corresponding user data, if authorized. Passive and active triggering methods are two different ways to find service provider or requester. The former happens when requested and the latter when a matchmaker agent finds proactively and automatically the service with using the user's profile and the ads from the service provider.

Human-to-Task agent (H2T)

Human-to-task agent (H2T) is an extended mode of H2H. When a user requests a service, the matchmaker recommends more than one task agent to perform a service for the user.

Human-to-Matchmaker (H2M)

Human-to-matchmaker (H2M) also is an extended mode of H2H. When a matchmaker receives a request from a user or user agent on behalf of the user and identifies that the request is out of the matchmaker's scope, it may introduce the user to another matchmaker and then relays the requests to the new matchmaker. H2M is to connect the user with a matchmaker which is more relevant to the user's request, mainly because the matchmaker could not be found in the vicinity of the user yet be found within the same zone. In comparison with H2M, M2M connects other matchmakers resided in other matchmaker community or zone.

Task agent-to-Task agent (T2T)

Task agent (TA) is used to perform the tasks issued by the user's request, to manage current service information. To do so, task agent notifies a priori its presence to the matchmaker in a continuous manner. If a task agent receives an out-of-capability request from a user, then the task agent could entrust the matchmaker with the user's task by having the matchmaker finds another appropriate task agent at hand. In other words, matchmaker may group a sort of task agents to let them cooperate with each other to come up with the user's requests.

Task agent-to-Matchmaker (T2M)

Task agent-to-matchmaker (T2M) is an extended mode of T2T. Task agent can collaborate with another task agent for giving a composed service in a seamless way to the user. If a matchmaker is connected to the task agent which has not information of the available task agents to co-work, the matchmaker may introduce a new matchmaker having information of appropriate task agent to cooperate in a vicinity to the task agent. The introduced matchmaker will receive information of collaborative task.

Matchmaker-to-Matchmaker (M2M)

In the complex real world, the matchmaker could be stressed out from lacking of capability to accept many or knotty user requests. This often happens when the interval time of the user requests is too short to response or accompanying with rich context data which are generated by complicated inferences. To improve rapidly increasing scalability, we can extend the capability of the matchmaker by using more than one matchmaker. In addition, M2M mode is needed when the requesters and resources to cope with the requests are too widely spread to be handled by one matchmaker. In special, zone-based group formation system should support this mode when the users allow the zone master to search for the most appropriate resource out of the zone. In this case, since the zone-based system employs its own matchmaker, the matchmaker is only accessible to the information of the resources within the zone. Therefore, when the matchmaker receives a user's requests over its capability, it will broadcast the user's requests to the matchmakers serving other zones. In this way, the linked matchmakers recommend the resources such as task agents or humans located in heterogeneous zones to cooperate with each other on behalf of the user's request. Despite its necessity, this kind of mode would have many things to consider such as security level and zone encapsulation, and hence be complicated to implement.

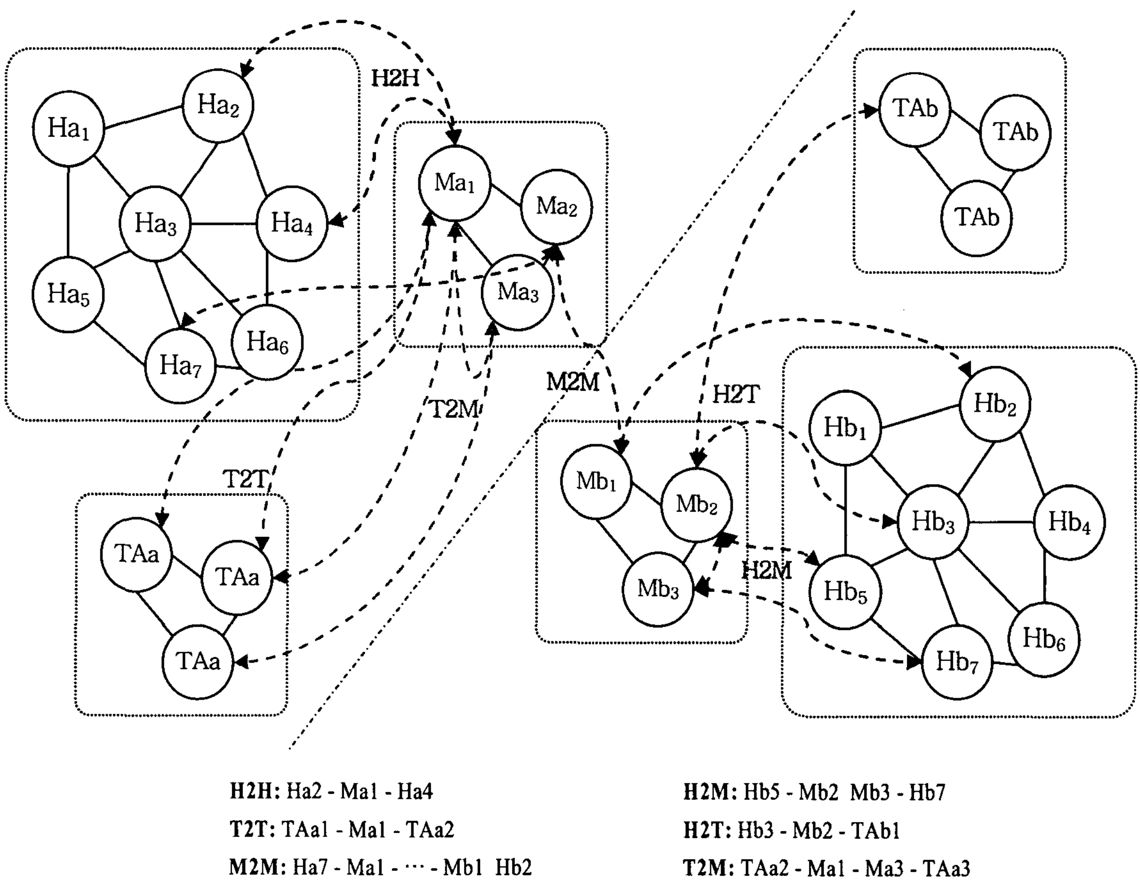
3.3 Matchmaking Knowledge Learning

We propose an ad-hoc group formation for the nature of a nomadic community. It means that it is difficult to acquire priori knowledge of domain experts so that rule-based system is not appropriate for the system. Hence, we employ case-based reasoning for the NAMA-US and propose refined case-based reasoning to extend differently from the classical case-based reasoning as follows:

Two methods are available to calculate for case-based reasoning: the nearest-neighbor retrieval and inductive retrieval. In the ubiquitous environment, people are moving actively so that the former is not appropriate for the system because it needs highly time-consuming calculation. Therefore, the latter is proper to NAMA-US, for relieving incomplete information problem, one of disadvantages of the latter, which causes difficulty to make a decision-tree, we propose several alternatives in the following.

Before introducing the alternatives against incomplete information, we find a couple of alternatives from prior research. Ströbel and Stolze proposed Extended Matchmaking Component (EMC) (Strbel and Stolze, 2002). The authors mentioned that if no constraints or only constraints from one side seller or buyer are defined, null-properties are the consequence. The authors suggest several alternatives to resolve these null-properties: (1) avoid situation, (2) domain constraints are required by the participants, (3) null-properties are marked as open issues and only properties with domain constraints are regarded, and (4) default properties are initialized which may include best practices or common trade standards.

The system should consider privacy issues which result in the tendency of avoiding to fully providing the information and missing context data. Therefore, we choose to improve the correctness through the process of default reasoning, mean-value, random selection, inference remaining features, and normalization after calculation the distance except missing features.



H2H: Ha2 - Ma1 - Ha4
T2T: TAa1 - Ma1 - TAa2
M2M: Ha7 - Ma1 - ··· - Mb1 Hb2
H2M: Hb5 - Mb2 Mb3 - Hb7
H2T: Hb3 - Mb2 - TAb1
T2M: TAa2 - Ma1 - Ma3 - TAa3

[Fig. 9] Classification of matchmaking mode

3.4 Triggering Method

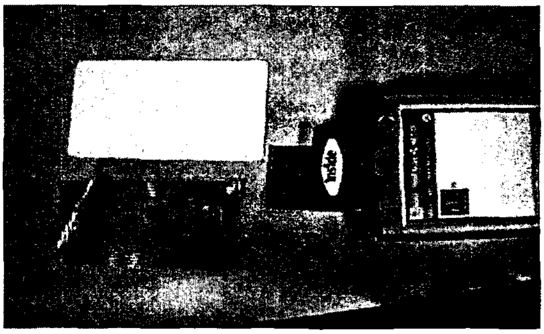
Due to from where to be triggered, the triggering type of matchmaking classifies into two methods: passive and active. The passive triggering method is a type of matchmaking which runs service discovery requested by service requester or service provider. In contrast to the former, the latter is an active type of matchmaking which the matchmaker gathers a user's profile and information of a task agent continuously so that the matchmaker provides appropriate service to each other proactively.

Ubiquitous computing should provide services proactively to users in an unconscious manner whether they ought to be served or not so that the system may not give any intrusion to the user to make them uncomfortable. Hence, the system should consider not only the passive matchmaking method required by users but also proper services to provide users by inferring the user's latent information and context data. Therefore, NAMA-US has designed to implement both active and passive matchmaking methods with using user ontology, service ontology, and task agent community and running multi-matchmaker agent all the time.

4. Implementation

Currently, since NAMA-US is an on-going project, the prototype system is not fully implemented. NAMA-US is a next-generation system of NAMA and NAMA-RFID. NAMA applies both Bluetooth and semantic web technologies to enhance context through user location tracking. To control the JSR-82 of Bluetooth, we use a JCP Linux-based Bluetooth controller API from Rococosoft Company, Impronto DK for Linux v1.1, the Windows operating system for the Pocket PC, and Atinav Company's aveLink Bluetooth Protocol Stack and SDK. To create the ontology, we used Protégé-2000; this was managed by an e-wallet web service using JENA 2.0 API. Pointbase was used to as the database running on the user's mobile device. For the Java virtual machine, Sun's JAVA SE 1.4.1 was used. For the Pocket PC virtual machine, EVM from Jeobe was applied. As a platform environment, Windows XP Professional and Linux-based Red Hat 9.0 was used. The mobile device operating system (in this case, two PDAs: HP's iPAQ 2100 and iPAQ 5400) is Microsoft Pocket PC 2003. We used a Bluetooth USB dongle with the Bluetooth v1.1 standard (PROMI-USB (Initium)).

NAMA-RFID uses RFID and semantic web based location-tracking method. To do so, ontology is managed by Protégé- and parsed by JENA 2.0. For PDA-based data management, a DB engine, Pointbase, is adopted. [Fig. 10] shows the RFID tag, reader and PDA used in the implementation.



[Fig. 10] RFID and PDA used in NAMA-US

5. Conclusion

This paper aims to illustrate how community computing technology and multi-agent-based web services can jointly contribute to a ubiquitous group formation support. To do so, an entire model of matchmaking modes is addressed for more structured approach to group formation system development. We selected and designed an ad-hoc group formation system as an example service. Afull-fledged context-aware service is considered: not only a location-based service but also a service that considers time, identity, location and entity.

To reveal the feasibility of the ideas in this paper, we proposed a RFID-based multi-agent-based group formation service called NAMA-US, which will run on the user's mobile device, such as a PDA. To automate the ad hoc group formation with user agent and GDSS task agent, we adopted semantic web with ontology.

Acknowledgments

This research is supported by the Ubiquitous Autonomic Computing and Network Project, the Ministry of Information and Communication (MIC) 21st Century Frontier R&D Program in Korea.

References

[1] Blake M. B., "The Business of e-Commerce Systems: Business Models and XML", 2002, *available from http://www.cs.georgetown.edu/~blakeb/Spring2002/COSC545/HTML_Pages/COSC545_Lecture5.pdf.*

[2] Cal, A., "IM3: A System for Matchmaking in Mobile Environments", *Lecture Notes in Computer Science*, Vol.3681(2005), 268-275.

[3] Carroll, J. M., "Community computing as human-computer interaction", *Behavior and Information technology*, Vol.20, No.5(2001), 307-314.

[4] Carroll, J. M. and M. B. Rosson, "Developing the Blacksburg Electronic Village", *Communications of the ACM*, Vol.39, No.12(1996), 69-74.

[5] Carroll, J. M., M. B. Rosson, P. L. Isenhour, C. H. Ganoe, D. R. Dunlap, J. Fogarty, W. A. Schafer, and Van C. A. Metre, "Designing Our Town: MOOsburg", *International Journal of Human-Computer Studies*, Vol.54(2001).

[6] Carroll, J. M., M. B. Rosson , P. L. Isenhour, Van C. Metre, W. A. Schaefer, and C. H. Ganoe, "MOOsburg: multi-user domain support for a community network", *Internet Research*, Vol. 11(2001), 65-73.

[7] Chun, K. J. and H. K. Park, "Examining the conflicting results of GDSS research", *Information & Management*, Vol.33(1998), 313-325.

[8] DeSantis, G. and Gallupe, R. B.. "A foundation for the study of group decision support systems," *Management Science*, Vol.33(1987), 589-609.

[9] Eiter, T., D. Veit, J. P. Mller, and M. Schneider, "Matchmaking for Structured Objects," *Lecture Notes in Computer Science*, Vol.2114 (2001), 186-194.

[10] Farrington, C. and E. Pine, *Community memory: a case study in community communication*, In P. Agre and D.Schuler (eds), Reinventing Technology, Rediscovering Community,

Mahwah, NJ: Ablex, 1997.

[11] Foner, L. N.m "Community Formation via a Distributed, Privacy-Protecting Matchmaking System," *Lecture Notes in Computer Science*, Vol.1519(1998), 359-376.

[12] Freuder, E. C. and R. J. Wallace, "Suggestion Strategies for Constraint-Based Matchmaker Agents", *Lecture Notes in Computer Science, Vol.1520(1998), 192-204.*

[13] Go, K. and J. M. Carroll, *Scenario-Based Task Analysis,* In Dan Diaper and Neville Stanton (Eds.), Handbook of Task Analysis for Human-Computer Interaction, 2003.

[14] Go, K., J. M. Carroll, and A. Imamiya, "Familyware: communicating with someone you love", *IFIP Conference on Information Technology at Home, Wolverhampton, UK, 28 - 30 June.In A.Sloane and F.van Rijn (eds), Home Informatics and Telematics: Information, Technology and Society (Boston, MA: Kluwer Academic)*, (2000), 125-140.

[15] Hamasaki, M. and H. Takeda, "Neighborhood Matchmaker Method: A Decentralized Optimization Algorithm for Personal Human Network", *Lecture Notes in Computer Science*, Vol.2773(2003), 929-935.

[16] He, M., N. R. Jennings, and H. F. Leung, "On Agent-Mediated Electronic Commerce", 2004, *available from http://citeseer.nj.nec.com/567141.html.*

[17] Hishiyama, R. and T. Ishida, "Modeling e-Procurement as Co-adaptive Matchmaking with Mutual Relevance Feedback," *Lecture Notes in Computer Science*, Vol.3371(2005), 67-80.

[18] Ishida, T., "Digital City Kyoto: Social Information Infrastructure for Everyday Life", *Communications of the ACM*, Vol.45, No.7 (2002), 76-81.

[19] Ishida, T., "Towards Communityware," *International Conference and Exhibition on the Practical Application of Intelligent Agents and Multi-Agent Technology (PAAM-97), Invited Talk*, (1997), 7-21.

[20] Kraemer, K. L. and J. L. King, "Computer-based systems for cooperative work and group decisionmaking", *ACM Computing Surveys*, Vol.20(1988), 115-146.

[21] Lee, K., W. Kim, and M. Kim, "Supply Chain Management Using Multi-agent System", *Lecture Notes in Computer Science*, Vol. 3191(2004), 215-225.

[22] Lu, H., "Mediator Based Open Multi-agent Architecture for Web Based Learning", *Lecture Notes in Computer Science*, Vol.2690 (2003), 94-98.

[23] McKillip, J., "Need analysis: Tools for the human service and education", *Applied Social Research Methods Series*, Vol.10(1987).

[24] Mynatt, E. D., A. Adler, M. Ito, and V. L. O'Day, "Design for network communities", *Proceedings of the ACM SIGCHI Confernece on Human Factors in Computer System*, 1997.

[25] Reviere, R., S. Berkowitz, C. C. Carter, and C. G. Gergusan, *Needs Assessment: A Creative and Practical Guide for Social Scientists*, Taylor and Francis: Washington, DC., 1996.

[26] Satoh, I., "Specification and Selection of Network Management Agents", *Lecture Notes in Computer Science*, Vol.2881(2003), 11-22.

[27] Shen, R., F. Yang, and P. Han, P. "A Dynamic Self-Organizing E-Learner Communities with Improved Multi-agent Matchmaking Algorithm", *Lecture Notes in Computer Science*, Vol.2903(2003), 590-600.

[28] Ströbel, M. and M. Stolze, "A Matchmaking Component for the Discovery of Agreement and

Negotiation Spaces in Electronic Markets", *Group Decision and Negotiation*, Vol.11(2002), 165-181.

[29] Sumi, Y. and K. Mase, "Collecting, Visualizing, and Exchanging Personal Interests and Experiences in Communities", *Lecture Notes in Computer Science*, Vol.2198(2001), 163- 174.

[30] Tesch, T. and P. Fankhauser, "Arbitration and Matchmaking for Agents with Conflicting Interests", *Lecture Notes in Computer Science*, Vol. 1652(1999), 323-334.

[31] Thibodeau, M., S. Bélanger, and C. Frasson, "WHITE RABBIT .Matchmaking of User Profiles Based on Discussion Analysis Using Intelligent Agents", *Lecture Notes in Computer Science*, Vol.1839(2000), 113-122.

[32] Van Dyke, N. W., H. Lieberman, and P. Maes, "Butterfly: A Conversation-Finding Agent for Internet Relay Chat", *IUI 99, Redondo Beach, CA, USA*., 1999.

[33] Vivacqua, A., M. Moreno, and J. Souza, "Profiling and Matchmaking Strategies in Support of Opportunistic Collaboration", *Lecture Notes in Computer Science*, Vol.2888 (2003), 162-177.

[34] Witkin, B. R. and J. W. Altschuld, *Planning and Conducting Needs Assessments: A Practical Guide*, Sage Publications: Thousand Oaks, CA., 1995.

[35] Wombacher, A. and B. Mahleko, "Finding Trading Partners to Establish Ad-hoc Business Processes", *Lecture Notes in Computer Science*, Vol.2519(2002), 339-355.

요약

유목커뮤니티 컴퓨팅에서 임의적 욕구파악과 그룹형성을 위한 욕구인지 다중에이전트 접근법

최근호* · 권오병*

최근에 커뮤니티 컴퓨팅에서 그룹형성과 그룹 의사결정에 관한 이슈가 제기되고 있다. 그러나 기존의 커뮤니티 컴퓨팅 시스템에서는 그룹의 욕구를 인지함에 있어서 임시적인 방법으로 그룹을 형성하지 못하고 있다. 임시적인 방법의 그룹형성은 유비쿼터스 컴퓨팅의 의사결정지원 시스템과 서비스에서 중요한 특성 가운데 하나이다. 따라서 본 연구에서는 유목 커뮤니티 컴퓨팅에서 임시적으로 욕구를 인지하고 그룹을 형성할 수 있는 NAMA-US 에이전트 중심의 다중 에이전트 방법론을 제시하고자 한다. 이 방법론은 어떤 커뮤니티에서 상대적으로 작은 그룹의 복수의 개별 사용자의 그룹의사결정을 지원할 때 다음과 같은 세 가지 특성을 만족하고자 한다. 첫째 임시적인 그룹형성과, 둘째 상황인지 그룹 욕구인지, 그리고 셋째 실내 · 외에서 동작 가능한 모바일 장비의 활용이다. RFID 기반의 프로토타입 시스템인 NAMA-US는 본 연구에서 제시하는 이런 개념을 실현시키기 위해 구축되었다.

Key words : 커뮤니티 컴퓨팅, 그룹 욕구 인식, 유비쿼터스 컴퓨팅, 임의적 그룹 형성, 다중에이전트 시스템

* 경희대학교 국제경영학과