Service-Oriented Autonomous Decentralized Community Communication Technique for a Complex Adaptive Information System

Khaled Ragab, Naohiro Kaji and Kinji Mori

Tokyo Institute of Technology
2-12-1 Ookayama, Meguro, Tokyo 152-8552, Japan
Tel: +81-3-5734-2664, Fax: +81-3-5734-2510,
Email: {ragab@mori., nkaji@mori, mori@}cs.titech.ac.jp

Abstract

In rapidly changing large-scale information system, users' requirements are changing constantly. To address the extreme dynamism in the large-scale information system, we have proposed the Autonomous Community Information System (ACIS). It is a decentralized bilateral-hierarchy architecture that forms a community of individual end-users (community members) having the same interests and demands in somewhere, at specified time. ACIS allows the community members to mutually cooperate and share information without loading up any single node excessively. In this paper, an autonomous decentralized community communication technique is proposed to assure a flexible, scalable and multilateral communication among the community members. The main ideas behind this communication technique are: content-code communication (community service-based) for flexible information service provision/utilization and multilateral benefits communication for scalable and productive cooperation among members. All members communicate productively for the satisfaction of all the community members. The scalability of the system's response time regardless of the number of the community members has been shown by simulation. Thus, the autonomous decentralized community communication technique reveals significant results of the response time with continuous increasing in the total number of members.

1. Introduction

The Internet is a highly heterogeneous and rapidly changing service environment [1]. It clearly conforms to Holland's description of complex adaptive systems: intrinsically dynamic, far from a global optimum, and continually adapting to new circumstances [2]. It promotes more sever and complex requirements for the information service systems and fosters an imperative need for high-assurance in these systems. These systems can be seen as instances of the *Complex Adaptive Systems* alike *social communities* [3].

Current Internet information systems are confronted to some challenges. First, the service providers (SP) provide information regardless of the end-users' demands and situations. There is no discernment between differences in place and time; end-users in any situation receive the same contents. Second, the number of worldwide Internet and mobile users are predicted to exceed 1 billion by the end of 2005 [4]. Those users have rapidly, and dynamically changing demands and interests. As a result, anywhere/somewhere at specified time there are significant numbers of users sharing the same interests and demands. Consequently a rapid and dramatic surge in the volume of requests arriving at a server often results in the server being overwhelmed and response times shooting up. Current information systems do not sustain such situation. For example, on the web the ubiquitous access of browsers and rapid spread of news about an event, lead to a *flash crowd* when a huge number of users simultaneously access a popular web site. Flash crowds are typically triggered by events of great interest, either planned ones such as sport events (e.g. FIFA 1998 world cup event [5]) or unplanned ones such as an earthquake, etc. However the trigger need not necessarily be an event of widespread global interest. Depending on the capacity of a server, even a humble flash crowd can overwhelm the server. Therefore, providing the well-customized service to the end-users is increasingly difficult, whereas end-users require well-customized, timely, continual, reliable, and available information services [6], [7]. Obviously, current Internet information systems have failed to fulfill the stringent Internet users' requirements in such situations [8]. Consequently there are increasing needs to design an assured information system that provides information that meets the users' demands.

To ride out the challenges above, we have been proposed the concept of an *Autonomous Community*

Information System (ACIS) [9]. It customizes the information service (e.g. news delivery) for the specific end-users (*community members*) have interests in that service, in somewhere/anywhere, at specified time. ACIS is completely decentralized in the sense that each member of the community performs the same set of tasks. It allows community members to communicate directly with one another and share information without relying on any specified servers. Community members mutually cooperate to assure the high quality and well-customized information service provision and utilization for all members.

The frequent join and leave of the community members sustain an increasing need for a flexible communication among them. The contribution of this paper is the proposition of the service-oriented community communication technique for achieving the flexibility and the timeliness in the complex adaptive information systems. The rest of this paper is organized as follow. Section 2 briefly presents the autonomous community information system concept and exhibits the system architecture. Section 3 exposes our proposed communication technique. Section 4 presents evaluation and simulation results showing improvement. Section 5 concludes this paper.

2. Autonomous Community Information System: Concept and Architecture

2.1 Concept

The main concern of the information systems has been in the past to efficiently retrieve relevant data for a particular request from immense repositories [10]. The research in information systems has turned to identify the location of the services and efficiently make the demands meet the offers [11]. In such distributed systems, two actors are coexisted: *Service Providers* and End-users. SPs offer the information content in the system. *End-users*, consume the information services.

In the information systems that have been designed on the basis of the centralized model, users individually send their requests to the service providers. Because of the end-users' demands are rapidly changing, in specified time the majority of them may have the same demands and then load up the service provider. For instance, the flash crowds caused by the September 11, 2001 terrorists attacked in the U.S. overwhelmed major news sites such as MSNBC and CNN, pushing site availability down close to 0% and response times to over 45 seconds [8]. Consequently, MSNBC quickly switched to serving static HTML and the percentage of error status codes dropped to 6.7%. Therefore, the information systems based on the centralized model are not scalable and failed to satisfy the Internet users' requirements of timeliness.

Currently, 90% of Internet resources are invisible and untapped [12]. *Peer-Peer information sharing systems* have turned to take into account the data and processing power that resides at the end-users. They drag information out of the centralized service providers onto end-users (peers) PC's. Peers share efforts for identifying the location of the required information. Then, information downloads are done directly between two peers [13]. These systems are characterized by *unilateral benefits* because of peers coordinate together for the satisfaction of only one of them, which requests the information. Thus, the average satisfaction rate for M peers in the systems is approximately 1/M (*reciprocal* relationship) and converges to zero as M increases. These systems have two lacks. First, the number of the identical requests is increased by the growth of the number of peers those send the same request. As a result a constant increase in traffic per peer is too high. In addition, peer-peer systems do not specify how many connections a peer may initiate, accept, or simultaneously maintain. Consequently some peers may have high load than others. Unfairness among users pushes them to give up from such systems. As a result the availability of the system is gradually decreased and the system becomes distasteful for its users. Obviously, these systems have failed to satisfy the Internet users' requirements (e.g. timeliness) too.

We have identified that the constructive cooperation among end-users assure the well-customized information service's provision and utilization. Blending the spirit of cooperation in the social communities, and the Autonomous Decentralized System (ADS) concept [14] [15], we have proposed the concept of *Autonomous Community Information System* (ACIS), [9]. We have defined *Autonomous Community* as a place where a coherent group of autonomous members having individual objectives, common interests and demands at specified time and somewhere/anywhere. The community members are autonomous, cooperative and active actors and they mutually cooperate to enhance the objectives for all of them timely and reliably. In ACIS, each community member acts both as an information sender and a receiver. Furthermore, each message from a participant is meaningful to all the other community members and at the same time every member is typically interested in data from all other senders in the community.

Contrary to the peer-peer systems, the communication among the community members is conducted on multilateral basis, as follows. Community members cooperate to locate of the requested information not only for the satisfaction for one of them that request the information but also for all of them. Thus, the average satisfaction rate for M members is approximately one. Moreover, as soon as a community member has new information, she/he publishes it to the

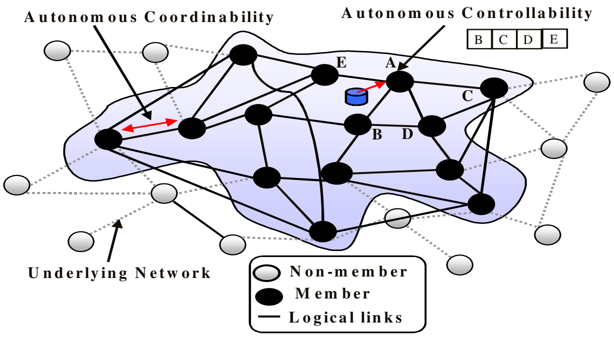

Fig. 1 Autonomous decentralized community system architecture.

whole community. Thus, the community members enrich their experiences and/or get to know new services without solicitation, in which individually they cannot get to know. As more and more members join the community, the information service is constantly scalable. In addition the availability of the ACIS gradually increases and becomes more attractive and useful for its members.

ACIS is a promising concept for information services operating at the edge of the network. It realizes the complex adaptive information system that successfully able to carry out, and enhance community members' objectives (e.g. timely information sharing) in a very dynamic environment. It guarantees the constructive cooperation and fairness among the community members with a very high degree of autonomy among them. We have developed a system architecture, called *Autonomous Decentralized Community System* (ADCS), that fosters the concept of the autonomous community information system.

2.2 Architecture

The autonomous decentralized community network is a self-organized logical topology. It is a set of nodes with considering the bilateral-hierarchy, the symmetric connectivity and the existence of loops. Community nodes are networked on a bilateral hierarchy basis. The bilateral logical contact between two community nodes will occur considering that the users of those nodes have same interests and demands, at specified time in somewhere. It is likely that in bilateral contacts, community members are get to know each other and share information. Each node keeps track of its immediate neighbors in a table contains their addresses. Each node knows its neighbor's nodes and shares this knowledge with other nodes for forming a loosely connected mass of nodes. For example, Figure 1 shows that each community node knows only four members. The bold lines represent the logical bilateral-link among the community nodes. Each node judges autonomously to join/leave the community network by creating/destroying its logical links with its neighbor's members based on its user's preferences.

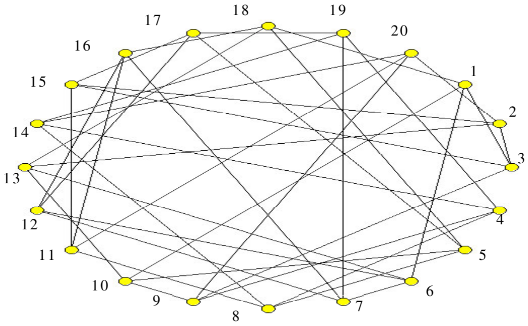

Fig. 2 Community network: Random-regular

2.2.1 Community Network Construction

Any node can join and leave the community at any time and via any node already in the community network. If no scheme is imposed on the way nodes join and leave the community network, then the network is likely to grow to become exponential network. This uncontrolled evolution may lead to some hotspots in the community network. For example, peer-peer systems do not specify how many connections a peer may initiate, accept, or simultaneously maintain. Consequently some peers may have high load than others. In that respect, we have proposed an autonomous decentralized community construction technique for making the potential hotspots very unlikely [9]. Community network construction polices the nodes joining and leaving the community network. Nodes are made to connect to randomly selected nodes already in the community network on joining with reserving the same degree for all the nodes and the short network diameter (cf. figure 2). Thus, the hotspots are very doubtfully and fairness is achieved in the community network by distributing the network traffic evenly among the community nodes during the communication.

2.2.2 Node Autonomy

Each node recognizes autonomously a member from a non-member and cooperatively forwards the community information to only its neighbor's members. Community node does not forward the community information/request out of the community. Moreover, each node "*think globally and act locally*" by taking a decision autonomously based on its local information to store the relevant received information. The decision is taken not only according to the node situation (e.g. limited resources) and the importance of the offered information but also according to the other members' requirements. Each community node keeps a short memory of the recently routed messages in order to avoid the congestion in the community network. Each node autonomously coordinates (cooperates) with the others for locating, and/or providing the information in the community.

If any member leaves, fails and joins the community the other community members still can coordinate their individual objectives among themselves. Consequently, each member is able to operate in a coordinated fashion.

ADCS architecture has no central server whatsoever, can be seen in figure 1. It is a fully decentralized model, where each participated node has equal responsibilities, and does not rely on any central authority to organize the network. Thus, it does not load up any single node excessively and enables the development of the large-scale information systems with adaptability, flexibility, reliability and high availability characteristics, which previously unseen in the Internet. For a scalable multilateral communication among the community members this paper proposes the technique that will be described in section 3.

3. Autonomous Decentralized Community Communication Technique

3.1 Conventional Communication

The conventional communication, typically through Web browsers, has been built on the one-to-one communication protocol. In one-to-one, data travels between two users, e.g., e-mail, e-talk. This protocol gobbles up the network bandwidth and makes the real time services unresponsive. Cashing most popular web pages on the proxy server reduces the network bandwidth consumption and the access latency for the users. However, the web cashes techniques have some disadvantages. First, a single proxy server is a single point of failure. Second, the limited number of users per proxy manifests bottleneck affects. Third, Data does not updated automatically. Finally, cash miss cause increases in the latency (i.e. extra proxy processing). While, in the conventional one-to-many group's communication the message travels primarily from a server to multiple users, e.g., web download and software distribution. For very large groups (thousands of members) or very dynamic multicast groups (frequent joins and leaves), having a single group controller might not scale well. Currently, there is no design for the application-level multicast protocol that scales to thousands of members. For example, *Overcast* [16] builds the mesh per group containing all the group members, and then constructs a spanning tree for each source to multicast information. The mesh creation algorithm assumes that all group members know one another and therefore, does not scale for large groups. *Bayeux* [17] builds a multicast tree per group. Each request to join a group is routed to a node acting as the root. This root keeps a list of all the group members. All group management traffic must go through that root. It generates more traffic for handling a very dynamic group membership. *Bayeux* ameliorates these problems by splitting the root into several replicas and partitioning members across them. But this only improves scalability by a small factor.

3.2 Service-oriented and Multilateral Community Communication Technique

Conventional communication techniques use the destination address (e.g. unicast address, multicast address) to send the data. In very changing environment likes ADCS, the state of the community nodes and the stability of connections are so unpredictable (i.e. end-users are frequently joined and left). Obviously, these conventional communication techniques are not applicable. Thus, the autonomous decentralized community communication technique has broached [18], to assure a productive cooperation and a flexible and timely communication among members. The main ideas behind our proposed communication technique are: content-code communication (community service-based) for flexibility and multilateral communication for timely and productive cooperation among members.

3.2.1 Service-oriented Community Communication

The first main idea behind the autonomous decentralized community communication technique is the separation of the logical community service's identifier from the physical node address. In this communication technique, the sender does not specify the destination address but only sends the content/request with its interest *content Code* (CC) to its neighbor's nodes. CC is assigned on a type of the community service basis and enables a service to act as a logical node appropriate for the community service. Figure 3 shows the community communication message format. CC is uniquely defined with respect to the common interest of the community members (e.g. politic, news, etc.). The information content is further specified by its *Characterized Code* (CH). The CH is the hash of the message content. It is uniquely specified with respect to the message content (e.g. data or request). We can compute it with the *collision resistance hash* function (e.g. SHA-1 [19]) that ensures a uniform distribution of CH.

CC	**CH**	**Data/Request**

Fig. 3 Community communication message format.

3.2.2 Multilateral Community Communication

The second main idea behind the autonomous decentralized community communication technique is *multilateral benefits* communication for timely and

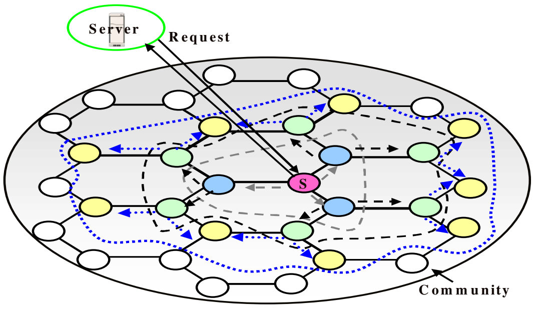

Fig. 4 Publish based protocol.

productive cooperation. The multilateral communication likely occurs among the community members that are already networked on a bilateral basis. All members communicate productively for the satisfaction for all the community members, as follow.

The autonomous decentralized community communication technique performs the communication among the community members that has called “*1→N*”. A brief scenario of the 1→N community communication is described as follows. The community node asynchronously sends a message to N neighbor’s nodes. Then, those N nodes forward the same message to another N nodes in the next layer, except the node that delivered the incoming message and so on gradually until all the community nodes received it. This technique handles, as the model knew like *viral propagation*. The autonomy of the 1→N communication can be seen as follow. Each community node recognizes autonomously member from non-member and judges autonomously to forward community messages to only N community neighbor’s nodes. In order to avoid the congestion that may be happening if some of the community nodes synchronously send identical messages, each node keeps a short memory of the recently routed messages and judges autonomously to forward only one copy of the received messages to the other neighbor’s nodes. Moreover, each node autonomously takes a decision to keep or delete the short memory of the received message based on the frequency of receiving such message.

The 1→N communication technique does not rely one any central controller. Each community node has its own local information and communicates only with specified number (N) of the neighbor’s nodes. There is no global information (e.g. multicast group address).

3.3 Community Communication Protocols

The autonomous decentralized community communication technique has two communication protocols: *publish based* and *request /reply-all based*.

- *Publish based protocol*. When one of the community members has new information, she/he publishes it to all the community members using “1→N”. The publish-based protocol offers an effective solution to the flash crowd problem as shown in figure 4. The solution scenario is as follows. As soon as one of the community members S has downloaded an interested content for the community from the server, she/he publishes it to all the community members, thereby relieving the server of this task and alleviating a load on the server. Thus, the load is distributed among the community nodes and increased slightly even as the number of nodes increases dramatically. In addition, it represents a scalable solution for large-scale information dissemination systems.

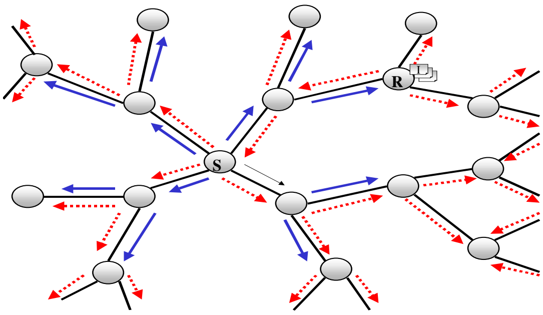

Fig. 5 Messages flow in request/replay-all based protocol.

- *Request/reply-all based protocol.* When a community member wants to locate information, she/he emits a request message. Then the others community members cooperate to locate the requested information. When any community node receives the requested message, it processes the request. If no results are found at that node, the node will forward the request to its neighbor’s nodes with using “1→N”. Otherwise, if any results are found at that node then the node will produce results, such as pointers to the information or the whole content based on the size of the information. Then that node will send a reply message not only to the node, which requested the information but also to all the community members. Figure 5 shows the message flow when the community node S sends a request (solid arrows) to its neighbor’s and node R replies (dotted arrows) to all the community members by the required information I. The *reply to all* protocol affluent the other community members to emit the same request. Consequently, all the community members enrich their experiences and/or get to know new services without requesting, in which individually they cannot get to know. Thus, the multilateral benefits characteristic of the community can be satisfied. In addition, it decreases the traffic per node by avoiding multiple requests for the same content.

The originality of our proposed communication technique does not come only from the *service-oriented* communication but also from the *reply-all* that satisfies the *multilateral benefits.* In 1→N community communication

Table 1: Comparison

			Conventional (client/server)	Peer-Peer	Community
Technological	Membership-Management		Centralized (Groupware)	Centralized (e.g., *Overcast, Bayeux)*	Decentralized Loosely control
	Communication	Model	Address-based	Address-based	Service-based
		Request	One-one	One-many	Cooperative(1→N)
		Reply	One-one	One-one	Cooperative(1→N)
		Feature	Unilateral	Unilateral	Multilateral
Characteristics	Users		Passive	Active	Active
	Load		Servers-congestions	Peers-congestions	No-Congestion (Fairness)

all members cooperate for the satisfaction of all the community members contrary to the peer-peer (P2P) communication techniques. In P2P systems, peers cooperate for the satisfaction of only one, which request the information (unilateral benefits). The comparisons between the community information system and the conventional information systems: client/server and peer-peer are tabulated in table 1. From this table we conclude that the community communication is: service-based, cooperative, relationship and multilateral communication. As more and more users join the community, the average satisfaction rate increases and the availability gradually increases. Moreover, the system is scalable of the response time with a huge number of members. Thus, it guarantees a timely communication among the community members.

4. Evaluation

To evaluate the performance of our proposed technique, we consider the community network topology has been constructed by our proposed construction technique [9] as random-regular graph. Thus, the number of the community nodes is

$$M \leq 1 + \sum_{i=0}^{L-1} k(k-1)^i \qquad (1)$$

Where k is the connectivity of each node and L is the diameter of the network (number of layers). Then the number of layers can be calculated as follows.

$$L \approx \left\lceil \frac{\log\left[\frac{(M-1)*(K-2)}{K}+1\right]}{\log(K-1)} \right\rceil \qquad (2)$$

Thus, the transmission time τ to send a message from one member to all the other members is bounded by O(Nlog$_N$(M)), where N=k-1.

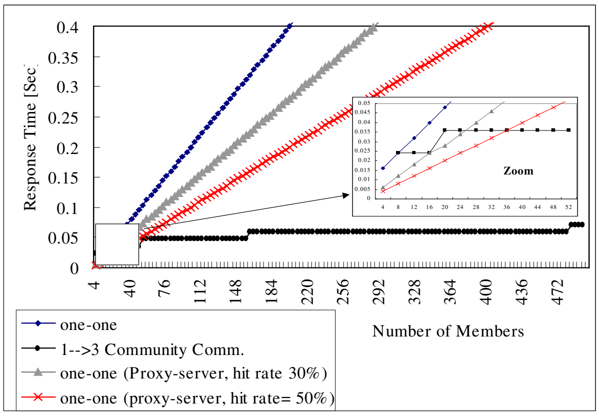

Fig. 6 Simulation result: scalable communication.

4.1 Simulation and Results

We simulated our communication technique on a network spending 4-array connectivity for each community node. Our experiment has been conducted over 100,000 community members, using 1→3 communication technique and is constituted of average communication cost between each node τ_{cc} = 0.001 seconds. Take τ_m = 0.001 seconds is the average time each nodes are needed for monitoring the recent received messages to avoid the congestion. Thus, the transmission time τ to send a message from any node to all the other community members is bounded by $L*N*(\tau_{cc}+\tau_m)$. For one-one communications, we assuming that all community members are from different organizations in the world so the web cashing techniques having slightly effect in the response time. It has been proven that a chasing proxy has an upper bound of 30-50% in its hit rate [20]. We tried our experiment for one-one communications with cashing proxy. We concentrate in this experiment on the comparison between the conventional one-to-one communication techniques without and with cashing proxy (hit rate of 30%, 50%) and (1→N) community communication technique. Figure 6 depicts the effectiveness of our communication technique in compared with the conventional ones. The 1→N communication technique is able to send a message to all the community members within an average of less than about 6 times in compare with the one-one communications. Furthermore, it shows that the community communication technique is scalable of the response time with the number of the members. For a very small number of members, our proposed community communication technique is not effective but it reveals great results with continuous increasing in the total number of members (see figure 6 zoom part).

5. Conclusion

Information service systems have radically altered the world of social and business and offer enormous potentials for e-social and e-commerce. Efficiently making the rapidly changing users demands meet the offer is of main importance in the large-scale and very dynamic information systems. While the current information technologies do not sustain the rapid and dramatic surge in the volume of requests arriving at a server. Thus, it is necessary to design a large-scale information system that meets the rapidly changing users' requirements for services with coping the extreme dynamism of the operating environment.

In this paper, we clarify the concept, architecture and communication technique. From the constructive cooperation in the social community and the ADS concept, we have proposed the autonomous community information system concept. In that respect, community members are active actors and the mutually cooperate to assure the quality of the information service provision and utilization among them, since individually they cannot. In addition, we have developed the bilateral-hierarchy system architecture, called *Autonomous Decentralized Community System* (ADCS) that sustains the proposed concept. Finally, we proposed the service-oriented autonomous decentralized community communication technique for achieving a productive cooperation and a flexible and timely communication among the community members. This communication technique is not only content-code communication (service-based) but also multilateral benefits communication. It allows all the community members to cooperate for the satisfaction of all members, contrary to the other communication techniques (e.g. peer-peer (P2P) communication). The simulation results has depicted and approved that the community communication technique is scalable of the response time with the number of the members. Moreover, the proposed communication technique shows an effective solution to the flash crowd problem.

We think that the request's pattern of information consumers determines an effect on the system. Determining the monitoring period with considering the request's pattern is our further research topic. In addition, we will investigate in the reliability for our communication technology.

6. Reference

[1] L. Introna and H. Nissenbaum, "Defining the web: The politics of search engines," IEEE Computer, vol. 33, no. 1, pp. 54-62, Jan. 2000.

[2] Holland, John H., "Using Classifier Systems to Study Adaptive Nonlinear Networks", in Lectures in the Science of Complexity, Santa Fe Inst. Studies in the Sciences of Complexity, Ed. D. Stein, Addison-Wesley Longman, 1989.

[3] C. G. Langton, "Complex Adaptive Systems," MIT Press, 1995.

[4] L. G. Roberts, "Beyond Moore's law: Internet growth trend ", IEEE Computer, Vol. 33, no. 1, p. 117, 2000.

[5] Martin Arlitt, Tai Jin, "Workload Characterization of the 1998 World Cup Web Site," Hewlett Packard Co. 1999.

[6] K. Mori, "Applications in Rapidly Changing Environments," IEEE Computers, Vol. 31, No. 4, PP. 42-44, 1998.

[7] David B. Ingham, S. K. Shrivastava, F. Panzieri, "Constructing Dependable Web Services," IEEE Internet Computing, Vol. 4, No. 1,pp. 25-33, 2000.

[8] J. Jung, B. Kirshanmurthy, and M. Rabinovich, "Flash Crowds and Denial of Service Attacks: Characterization and Implications for CDNs and Web Sites", The 11th WWW2002 Conference, Hawaii, USA, May, 2002.

[9] K. Ragab, T. Ono, N. Kaji, K. Mori, " Autonomous Decentralized Community Concept and Architecture for a Complex Adaptive Information System," Proc. IEEE FTDCS, Puerto Rico, May 2003.

[10] V.N. Gudivada, V.V. Raghavan, W.I. Grosky, and R. Kasanagottu, "Information retrieval in the World Wide Web", IEEE Internet Computing Magazine, Vol. 1, no. 5, pp. 58-68, Sept. Oct. 1997.

[11] K. Mori, "Assurance System Architecture for Information Service by Utilizing Autonomous Mobile Agents", Fifth IEEE International High-Assurance Systems Engineering Symposium, Nov. 2000.

[12] R. Dornfest, "Dark Matter, Sheep and the Cluster: Resolving Metaphor Collision in P2P," The O'Reilly Peer-to-Peer and Web Services Conference Washington, D.C., November 5-8, 2001

[13] M. Ripeanu, "Peer-Peer Architecture Case Study: Gnutella Network," In Proc. of Int. Conf. on P2P computing, 2001.

[14] K. Mori, "Autonomous Decentralized Systems: Concept, Data Field Architecture and Future Trends," Proc. of the first Int. Sym. On ADS, (ISADS'93), IEEE, Kawasaki, Japan, pp. 28-34, 1993.

[15] K. Mori, H. Ihara, et al., "Autonomous Decentralized Software Structure and its Application," Proc. IEEE FJCC'86, pp.1056-1063. November 1986.

[16] J. Jannotti, D. K. Gifford, K. L. Johnson, M. F. Kaashoek, and J. W. O'toole. "Overcast: Reliable Multicasting with as n Overlay Network," In Proc. Of the 4th Symposium on (OSDI), pages 197-212, Oct. 2000.

[17] S. Q. Zhuang, B. Y. Zhao, A. D. Hoseph, R. H. Katz and J. D. Kubiatowicz, "Bayeux: An Architecture for Scalable and Fault tolerant Wide-area Data Dissemination," In Proc. Of the 11th Int. (NOSSDAV) Workshop, June 2001.

[18] K. Ragab, T. Ono, N. Kaji, K. Mori, "Community Communication Technique for Achieving Timeliness in Autonomous Decentralized Community Systems," Proc. IEEE IWADS, Beijing, China, pp 56-60, Nov., 2002.

[19] FIPS 180-1 Secure hash standard. Technical Report Publication 180-1, Federal Info. Processing Standard (FIPS), National Inst. Of Standards and Technology, US Dept of Commerce, Washington D.C., April 1995.

[20] Stephen Williams, and Edward A. Fox. "*Caching proxies: Limitations and potentials,*" In Proceedings of the Fourth International World Wide Web Conference, December 1995.