

Autonomous Decentralized Community Concept and Architecture for a Complex Adaptive Information System

Khaled Ragab, Takanori Ono, Naohiro Kaji and Kinji Mori

Tokyo Institute of Technology

2-12-1 Ookayama, Meguro, Tokyo 152-8552

Tel: +81-3-5734-2664, Fax: +81-3-5734-2510,

Email: fragab@mori., nouri@mori., nkaji@mori, mori@cs.titech.ac.jp

Abstract

The global information service in the Internet is a heterogeneous and rapidly evolving environment. Constantly, new information services are added, others are modified, removed or in fault, making it more and more intractable to maintain a coherent image of the information environment. Moreover, users' interests and demands for information services are rapidly changing. In this paper we propose the concept of an Autonomous Community Information System (ACIS) to meet the users' requirements and guarantees the evolution and continuity of the information systems. It allows individual end-users (community members) to communicate directly with one another and share information without relying on any centralized authorities to organize the network. Moreover, it does not load up any single node excessively.

For an efficient communication among the community members, we propose an autonomous decentralized community construction technique. It makes hotspots in the community network very improbably and satisfies the fairness by distributing the network traffic evenly among the community members. In addition, it is highly scalable because the complexity of this technique at each node grows logarithmically with the number of the community members.

1. Introduction

The Internet's phenomenal impact, the subsequent growth and the evolving in social and economic environments raise more sever and complex requirements for the information service systems. The information systems are confronted to two challenges. First, the number of worldwide Internet users is increasing very quickly with a grow rate of 20% annually [1] and it is predicted to exceed one billion by the end of 2005. In addition, the world

publishes on-line about 300 terabytes of information every year [2]. It reflects the extremely large size of the Internet that contains more than 800 million [3] pages of various types. As more and more users and information engaged in the Internet, users are finding themselves increasingly overwhelmed with the available information, and no single centralized systems or portals can timely locate, and catalog ever-increasing information. As a result, locating the required information in real time is increasingly difficult. Currently, the search engines index about only 25% of the available services to the users [4], [5]. We believe that increasing information system complexity shall reach a level beyond human ability to manage and maintain.

Users require continuous, reliable, and available information service [6]. Under the evolving situations, they have heterogeneous and dynamically changing requirement levels of timeliness, and reliability [7]. Timeliness and reliability are essential components in modern high assurance systems [8]. Obviously, current information systems have failed to fulfill the stringent Internet users' requirements. Consequently there are increasing needs to design information systems that characterized by decentralized control, large scale and extreme dynamism of their operating environments to meet the users' heterogeneous demands. These systems can be seen as instances of Complex Adaptive Systems. Social community is an instance of complex adaptive systems [9]. Cooperation is the key of evolution and continuity of the social communities. The potential benefits of cooperation among people should drive progress evolution in culture, technologies, and business [10]. Inspired from both the spirit of cooperation in the social communities, and the Autonomous Decentralized System (ADS) concept [11] [12], the concept of *Autonomous Community Information System (ACIS)* is proposed to meet the users' requirements and guarantee the evolution and continuity of the

information systems. We propose ACIS to allow individual end-users to communicate directly with one another and to share information without relying on any centralized server. It is completely decentralized: each member of the community performs the same set of tasks and serves all the other community members.

In the client-server model, the value of the network to individual users decreases with the dramatically increasing of the total number of participants in the network. In this paper we propose an autonomous decentralized community construction technique to exhibit what economists call the "network effect" [13] in which the value of the community to a community member scales with the total number of participants. The contribution of this paper is the proposition of the autonomous community information system concept and architecture for a complex adaptive information system. In addition the proposition of the autonomous decentralized community construction technique. Consequently, our proposed construction technique achieves fairness among community members.

The remainder of this paper is organized as follows. Section 2 clarifies the autonomous community information system concept. Section 3 exhibits the system architecture and the autonomous decentralized community communication technique. Section 4 exposes our proposed autonomous decentralized community construction technique. The last section draws conclusions.

2. Autonomous Community Information System Concept

The main concern of the information systems has been in the past to retrieve efficiently relevant data for a particular request from immense repositories [14]. The research in information systems has turned to identify the location of the services and efficiently make the demands meet the offers. In such distributed systems two actors are coexisted:

- **Service Providers**, offer the information content in the system.
- **End-users**, consume the information services.

The current information systems have been designed on the basis of the centralized Model. They did not take into accounting the data and processing power that resides at the end-users. The PC's of these users are the dark matter of the Internet, i.e. 90% of the Internet resources are invisible [15]. In such systems, users are passive actors (receive, but do not contribute, information) and their PC's are nothing more than a life-support system for a browser and a place to store cookies. As the number of users and information, are rapidly growths service providers become increasingly

overloaded. For instance, the flash crowds caused by the September 11, 2001 terrorist attacks 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 [16]. In addition, MSNBC quickly switched to serving static HTML and the percentage of error status codes dropped to 6.7%. Thus the current information systems have failed to satisfy the Internet users requirements. Obviously, end-users collectively and cooperatively have more significant data and processing power than centralized service providers. Thus the next generations of the information systems must be characterized by large scale, decentralized and cooperative management. They will reverse today's "everything onto the Web!" trend and drag information out of the centralized service providers onto end-users PC's. They must bring a new consequence to the end-users and make them active actors. The cooperation among users is the key of the existence, evolution and continuity of such systems. The community as a social phenomenon deals with establishing and working with meaningful connections between people. The constructive cooperation among them guarantees the successful community. The value of the individual is enhanced, because many systems emphasize each person's unique contributions. However, the value of individual differences does not lead to rampant individualism. Instead, member-to-member raises the value of the community. There is no such thing as a single individual. The value of a member grows together with his ties to all the other members in the system. Thus, the member gains value insofar as he or she contributes to the community and obtains the community's benefits.

Under the dynamically changing environments, the system has to cope with the unsteady network conditions, and extreme dynamism in users' demands and interests. The system has consequently to assure the online-expansion, maintenance and fault tolerance of the information systems. Inspired from both Autonomous Decentralized System (ADS) concept [11] [12], and the cooperation in the social communities, we propose the concept of *Autonomous Community Information System* (ACIS). In fact there are large numbers of articles have given definitions of the community from the social point of view. Inspired from the ADS concept, we define *Autonomous Community* as a place where a group of autonomous members having individual objectives, common interests and mutually cooperate to enhance their objectives. We used the term community as a metaphor of the new wave of computing technologies. ACIS has the following properties:

- *Autonomy*. The actors of the system are members. They autonomously manage themselves and

autonomously coordinate with the others.

- *Adaptability*. The system is self-adapting to the change of the members' interests and demands.
- *Synergy*. Mutual cooperation improves the performance of the individual members and increases the availability of the system they form. As more and more members participate in the community, the system is constantly scalable.

It is a promising concept for information services operating at the edge of the network. It realizes an adaptable information sharing system that successfully able to carry out, and enhance community members' objectives (information sharing) in dynamic environments. It doesn't guarantees only the autonomy for each member but also the cooperation and fairness among members. Members are active actors in providing and utilizing the information as well. Community brings new importance to end-users. Moreover, the community members improve their objectives by mutually support and can do things that individually cannot. For instance, in the autonomous community information systems, members cooperatively locate their requested information timely and reliably. We have developed a system architecture, called *Autonomous Decentralized Community Information System (ADCIS)*, that fosters the concept of autonomous community information system.

3. Autonomous Decentralized Community Information System Architecture

The autonomous decentralized community network is a self-organized network. It is a set of nodes with considering the non-hierarchy, symmetric connectivity and the existence of loops. Each node keeps track of its immediate neighbors in a table contains their addresses. Each node knows its neighbor nodes and shares this knowledge with other nodes to form 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 link among the community nodes. Each node judges autonomously to join/leave the community topology by creating/destroying its logical links with its neighbor's members based on its user's preferences. Section 4 presents our proposed technique to construct community network. Each node recognizes autonomously 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 judges autonomously to store the relevant received information according to its situation and the importance of the offered

information. Each community node keeps a short memory of the recently routed messages 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 left the community, his node failed, or a new member joined the community, the other community members still can coordinate their individual objectives among themselves and each member able to operate in a coordinated fashion. The community boundary changes with the dynamic change of its members' requirements.

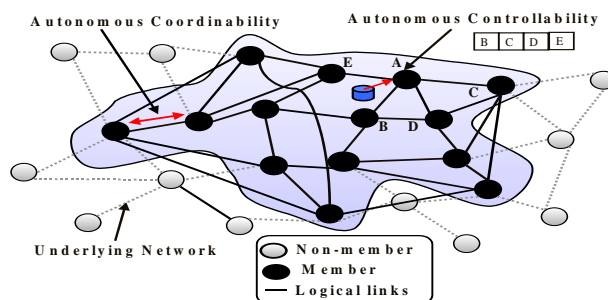


Figure 1. Autonomous decentralized community architecture

ADCIS architecture has no central server whatsoever, as you can see in Figure 1. The autonomous community network is constructed as follows. The end-user who wants to join a community has to discover at least one of the community nodes on the network. Then he interacts with each of them for sending and receiving the community contents. The end-users' nodes can either use information from a local configuration scheme to discover a community node (for example, a configuration entry that tells it who to talk with) or it can employ network broadcasting and discovery techniques such as IP multicast to discover the other community members. The strength of this architecture is that it does not rely on any one server to be available for registration of its location in order for other members' node to find it. ADCIS architecture is fully decentralized model, where each participating node has an equal's responsibilities. It does not rely on any central authority to organize the network and does not load up any single node excessively. It enables the development of the information systems with adaptability, reliability and high availability characteristics previously unseen in the Internet. For a real-time communication among the community members we proposed the autonomous decentralized communication technique in section 3.1.

3.1 Autonomous Decentralized Community Communication Technique

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 clients, e.g., e-mail, e-talk. This unicasting protocol gobbles up the network bandwidth and makes the real time services unresponsive. Caching most popular web pages on the proxy server reduces the network bandwidth consumption and the access latency for the clients. In contrast, the web caching techniques have some disadvantages. First, a single proxy server is a single point of failure. Second, the limited number of clients per proxy manifests bottleneck affects. Third, Data does not updated automatically. Finally, cash miss cause increase in latency (i.e. extra proxy processing). In the conventional one-to-many group communication the message travels primarily from a server to multiple clients, 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 application-level multicast protocol that scales to thousands of members. *Overcast* [17] builds the mesh per a group containing all 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* [18] builds a multicast tree per a 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. It generates more traffic for handling a very dynamic group membership. In particular, all group management traffic must go through the root. *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.

ADCIS is a very dynamic system: users are frequently joined and left. As a result handling ever-change community membership on a central authority is impossible. In [3], an autonomous decentralized community communication technique has been broached. It performs the communication among the community members and we call it "1->N". Unlike the conventional communications are address-based, our proposed communication technology is content-code based. A brief scenario for the 1->N community communication is 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 and so on gradually layer by layer until all the community nodes received the message. This community communication technique handles, as the model knew like *viral propagation*. It broadcasts the message gradually layer by layer to all the community nodes. It does not rely one any central controller. Each community node has its own local information and

communicates only with specified number (N) of neighbor's nodes. There is no global information (e.g. multicast group address). Autonomous decentralized community is *loosely connected* mass of nodes and *loosely controlled*. Autonomous decentralized community communication technique has the two communication protocols: *publish-based broadcast*, and *request-based broadcast*.

- When one of the community members has new information she/he broadcasts it to all the community members using "1->N". Each node autonomously judges to store the relevant received information according to its situation (e.g. limited resources) and the importance of offered information. The publish-based communication protocol offers an effective solution to the flash crowd problem. The flash crowds are typically triggered by events of vast interests, whether planned ones such as 1998 FIFA World Cup event [20] or unplanned ones such as a plane crash or such as an earthquake. The solution scenario is as follows. As soon as one of the community members downloaded an interested content for the community from the server, he/she publishes it to all the community members, thereby relieving the server of this task and alleviating 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 dissemination systems.
- When a new node joins the community and emits a request message, the others community nodes cooperate to locate the requested information. When any community node receives a request 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 information) and the node will send a single reply message back to the requester via the reverse path traveled by the request. The reply by reverse path technique emerges weakness when very large number of nodes joins the community and those nodes emit the same request. As a result a constant increase in traffic per node is too high. Because of the autonomous community is loosely controlled and no community node knows total information about the system. So we select to reply to all the community nodes using "1->N" instead of reply by reverse path to the requester only. The reply to all technique affluent the other community members to emit the same request.

This communication technique is characterized by

bilateral benefits. In other word all members cooperate for the satisfaction of all the community members. As more and more users join the community, the availability gradually increases and becomes more attractive and useful to all its members. Moreover, it guarantees fairly, uniformly and timely communication among the community members. For efficient broadcasting, we propose an autonomous decentralized community network construction technique in section 4.

4. Autonomous Decentralized Community Construction Technique

4.1 Goals and Requirements

The autonomous decentralized community network is to be symmetric in the sense that each node in the network is to have identical capabilities and duties on the network. This excludes the existence of central servers that might be involved in organizing the network. Our first goal is to construct a community network that can support broadcast efficiently. In addition, we should avoid hotspots in the community network by distributing the network traffic evenly among the community nodes during the broadcast. Fairness among community nodes is highly required. Unfairness among members pushes them to give up from the community system, as a result the availability of the system is gradually decreased and the system becomes distasteful for its users. The fairness objective dictates that the all nodes must have the same degree as possible. The second goal is to make our community network construction is highly scalable. In other words, the complexities for joining and leaving nodes should be as low as possible. Finally, the community network topology has to provide redundancy. Node failures must not lead to the community network disconnecting or severely hampering broadcast properties. Next subsection presents our proposed construction technique for the community topology.

4.2. Organizing nodes in a Random Graph

It has been observed that the node degree distribution of the Internet with ‘uncontrolled evolution’ mimics a power-law. It occurs when no scheme is imposed on the way nodes join and leave the network. Any node can join and leave the network at any time and via any node already in the network. Such networks are likely to grow to become exponential networks. This uncontrolled evolution may lead to some hotspots in the network. In that respect, there are increasing needs to construct the community network with making the potential hotspots very unlikely. Thus, we

have decided to construct community network with a randomized technique. In this paper we construct the community network as regular multi-graphs that are composed of independent *Hamilton* cycles. We construct it as regular random multi-graph for three arguments as follow. First, regular graphs are chosen because it is required that all nodes having the same degree. Second, we construct the community network as an intermediate of the completely ordered regular network and the fully random network for achieving two interesting features: high clustering i.e., there is a high density of connections between nearby nodes, which is a characteristic of the regular topologies, and short network diameter, which is typically observed in random topologies. Finally, the community network composed of Hamilton cycles having the advantage that nodes joining or leaving will only require local changes in the graph. Consider a multi-graph $G = (V, E)$, such that V is the set of nodes with labels $[n] = \{1, 2, \dots, n\}$ and E is the set of edges. Assume H_n denote the set of all Hamilton cycles on set $[n]$, and $C_1, C_2, \dots, C_d \in H_n$ for $n \geq 3$, $E = (C_1, C_2, \dots, C_d)$. Let $H_{n,2d}$ be the set of all $2d$ -regular multi-graph constructed by some $C_1, C_2, \dots, C_d \in H_n$. To construct the community network with achieving the goals mentioned before, we shall construct $2d$ -regular multi-graphs in $H_{n,2d}$ for $d \geq 2$ and $n \geq 3$. Of course if $n \leq 2d + 1$ we might use a complete graph instead of H -graphs. For each node the neighbors are labeled as $r_p^{(1)}, r_s^{(1)}, r_p^{(2)}, r_s^{(2)}, \dots, r_p^{(d)}, r_s^{(d)}$. For each i , $r_p^{(i)}$ denote the neighbor node’s predecessor and $r_s^{(i)}$ denote the neighbor node’s successor on the i th Hamilton cycle.

4.2.1 Discovering Community Network

The end-user who wants to join a community has to discover at least one of the community nodes. The end-user’s node can either use information from a local configuration scheme to discover a community node (for example, a configuration entry that tells it who to talk with) or it can employ network broadcasting and discovery techniques such as IP multicast to discover the other community members. The strength of this approach is that it does not rely on any one server to be available for registration of its location in order for other member’s node to find it.

4.2.2. Member Joining/Leaving Process

As soon as the end-user’s node discovered one of the community nodes, it sends a join request to this node. In this paper, we propose two basic operations for the community network construction: *Join* and *Leave*. The *Join(x)* operation inserts a new node x into the community network graph G . Any node in the graph G must be able to accept a join request at any time. In addition, any node in G can also

call the *Leave()* operation to remove itself from the graph G. The operations pseudo codes are tabulated in table 1.

<pre> Edge (x, y, i){ (x ⇒ r_s⁽ⁱ⁾) ← y (y ⇒ r_p⁽ⁱ⁾) ← x } </pre>
<pre> Add (x, i){ Edge (Calling_node, x, i) Edge (x, Calling_node ⇒ r_s⁽ⁱ⁾, i) } </pre>
<pre> Join (x){ For i :=1, ..., d in parallel do z_i := Randomize() z_i ⇒ Add(x, i) } </pre>
<pre> Leave () { For i :=1, ..., d in parallel do Edge (r_p⁽ⁱ⁾, r_s⁽ⁱ⁾, i) } </pre>

Table 1. Operations pseudo codes

The expression $h \Rightarrow \text{var}$ means that we seek the value of var from node h. In addition, the expression $h \Rightarrow \text{Proc}()$ calls a remote procedure Proc at node h. Finally, $(h \Rightarrow \text{var}) \leftarrow y$ means that we set the variable var of node h to value y. The procedure *Edge*(x, y, i) connects two nodes x and y on the level-i cycle, and procedure *Add*(x, i) makes node x the successor of the calling node. A new node x joins by calling *Join*(x) at any existing node in the graph. Node x will be inserted into the cycle i between the node z_i and the successor of the node z_i ($z_i \Rightarrow r_s^{(i)}$) for a randomly chosen z_i 's for $i=1, \dots, d$. Figure 2 shows the joining operation for the new node F. For clarity we show only two Hamilton cycles C_1 and C_2 . The *Randomize()* function returns a node that was chosen uniformly at random from the graph as follows. A new node x broadcasts a request to all nodes. Each node will be asked to reply with a certain probability $p = O(1/n)$. Thus node x expects to receive some replies and can autonomously select randomly from them. Clearly as n increases the probability p for the new joining node to connect to the previous selected one is converged to zero ($\lim_{n \rightarrow \infty} p = 0$). Moreover, a broadcast on a H-graph sends at most $(2d-1)n$ messages and terminates $O(\log_d n)$ steps with high probability.

The information on the topology of the community network is packaged in this technique. It is used to police nodes joining and leaving the community network: Instead of allowing nodes, to join and depart without any restrictions, 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. The constructed topology is an intermediate of the completely ordered regular topology and the fully random

topology.

In this paper we construct the community network as random regular graph instead of completely ordered regular graph. The simple type of the completely ordered regular graph consists of a ring of n vertices, each of which connected to its nearest k neighbors. For example if k is 8, each vertex is connected to its nearest four neighbors on each side (eight in total). When n is much larger than k ($k \ll n$) then the diameter of this graph can be show approximately as $n/2k$. For example, if n is 8,192 and k is 8, then $n/2k$ is 512- a very large number of hops to get the opposite side of the ring. While the random regular graph in $H_{n,2d}$ ($k=2d$), vertices are connected to one another in random. For a large random regular graph the diameter is approximately $\log_k n$. So using our previous example where n was 8,192, and k was 8, then diameter would be $\log_8 8,192 = 4$ - much better than 512 hops for the completely ordered graph. Figure 3, shows that the community network diameter grows logarithmically with the network size.

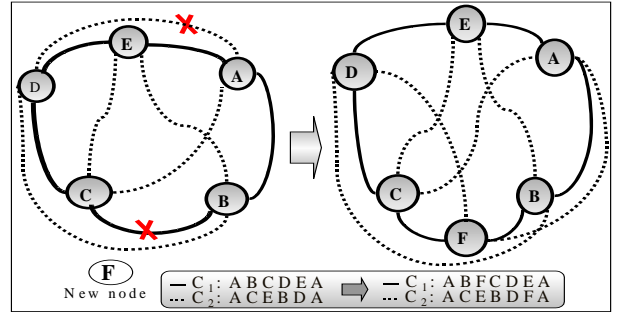


Figure 2. Joining operation

In [21] authors have proposed an algorithm for building low-diameter peer-peer networks with bounded degrees. Their approach and techniques are different with ours. Their protocol relies in one host server, which maintains knowledge of the global topology. This server gives the new joining node D random nodes to contact. Thus this approach is not scalable for a large number of users. Contrary, our protocol is a fully decentralized one for membership management. Thus, the autonomous decentralized community system is not only *loosely connected* mass of nodes as we mentioned before but also *loosely controlled*. We conclude that with forming the community network as a regular-random graph, allows message to spread much more rapidly across the community network. Thus our proposed community network construction assures more efficient communication technology. Now we are working for simulating the proposed technology for joining and leaving.

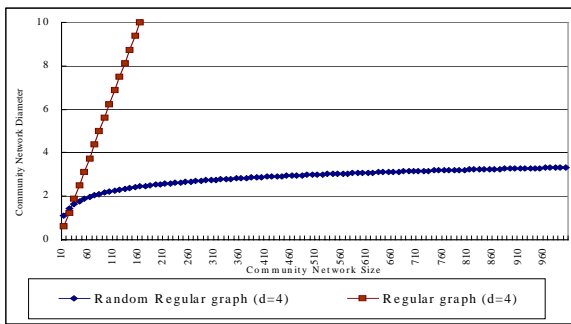


Figure 3. Network diameter growth logarithmically

5. Conclusion

The global information services are rapidly evolving and the users' requirements for information services are change quickly. Moreover, the complexity of information system increases and shall reach a level beyond human ability to manage. The current information systems are failed to satisfy the ever change and stringent Internet users' requirements timely. As a result the information systems should be characterized by decentralized control, large scale and extreme dynamism of their operating environment. In this paper, we have developed a system architecture, called *Autonomous Decentralized Community Information System (ADCIS)*, that fosters the concept of autonomous community information system. It realizes the adaptable information sharing system that successfully able to carry out, and enhance community members objectives (information sharing) in dynamic environments.

For efficient communication among community members, we propose an autonomous decentralized community network construction technique. It is completely decentralized technique. It supports broadcasting efficiently. In addition, it makes the hotspots in the community network very doubtfully and achieves fairness by distributing the network traffic evenly among community nodes during broadcast. Moreover it is highly scalable because the complexity of joining is $O(\log_d n)$ with $O(d \log_d n)$ messages and the complexity of leaving is $O(1)$, with $O(d)$ messages.

5. References

[1] L. G. Roberts, "Beyond Moore's law: Internet growth trend", IEEE Computer, Vol. 33, no. 1, p. 117, 2000.
 [2] P. Lyman and H. R. Varian, "How Much Information", Journal of Electronic Publishing, Vol. 6, Issue 2, Dec. 2000.
 [3] L. Inrona and H. Nissenbaum, "Defining the web: The politics of search engines", IEEE Computer, vol. 33, no. 1, pp. 54-62, Jan. 2000.

[4] R. Hunt, "Evolving technologies for new Internet applications", IEEE Internet Computing Magazine, Vol. 3, no. 5, pp. 16-26, Sept.-Oct. 1999.
 [5] S. Lawrence and L. Giles, "Accessibility and distribution of information on the web", Nature, Vol. 400, pp. 107-109, July 1999.
 [6] D. B. Ingham, S. K. Shrivastava, F. Panzieri, "Constructing Dependable Web Services," IEEE Internet Computing, Vol. 4, No. 1, pp. 25-33, 2000.
 [7] K. Mori, "Applications in Rapidly Changing Environments," IEEE Computers, Vol. 31, No. 4, PP. 42-44, 1998.
 [8] K. Mori, "Towards integrated methods for high assurance systems," IEEE Computer, vol. 31, No. 4, PP. 32-34, 1998
 [9] C. G. Langton, "Complex Adaptive Systems", MIT Press, 1995.
 [10] John Stewart, "Evolution's Arrow: The direction of evolution and the future of the humanity", Chapman press, Australia, 2000.
 [11] 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.
 [12] K. Mori, H. Ihara, et al., "Autonomous Decentralized Software Structure and its Application," Proc. IEEE FJCC'86, pp.1056-1063. November 1986.
 [13] M. Katz and C. Shapiro, "Systems Competition and Network Effects", Journal of Economic Perspectives, Vol. 8, no. 2 pp. 93-115, 1994.
 [14] V.N. Gudivada, V.V. Raghavan, W.I. Grosky, and R. Kananagottu, "Information retrieval in the World Wide Web", IEEE Internet Computing Magazine, Vol. 1, no. 5, pp. 58-68, Sept. Oct. 1997.
 [15] 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., Nov. 5-8, 2001
 [16] J. Jung, B. Kirshanmurthy, and M. Rabinovich, "Flash Crowds and Denial of Service Attacks: Characterization and Implications for CDNs and Web Sites", The 11th Int. World Wide Web Conf., WWW2002, Hawaii, USA, May, 2002
 [17] 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 Fourth Symposium on Operating System Design and Implementation (OSDI), pages 197-212, Oct. 2000.
 [18] S. Q. Zhuang, B. Y. Zhao, A. D. Hoseph, R. H. Katz and J. D. Kubiawicz, "Bayeux: An Architecture for Scalable and Fault tolerant Wide-area Data Dissemination", In Proc. Of the 11th Int. workshop on Network and Operating Sys. Support for Digital Audio and Video (NOSSDAV 2001), June 2001.
 [19] K. Ragab, T. Ono, N. Kaji, K. Mori, "Community Communication Technology for Achieving Timeliness in Autonomous Decentralized Community Systems", IWADS, Beijing, China, Nov., 2002.
 [20] Martin Arlitt, Tai Jin, "Workload Characterization of the 1998 World Cup Web Site", Hewlett Packard Co. 1999.
 [21] G. Pandurangan, P. Raghavan, E. Upfal, "Building Low-Diameter P2P Networks", In Proc. Of the 42nd Annual IEEE Symp. On the foundations of Computer Science, Oct., 2001.